Національний університет «Полтавська політехніка імені Юрія Кондратюка»

(повне найменування вищого навчального закладу)

\_Навчально-науковий інститут інформаційних технологій та робототехніки\_

(повна назва факультету)

\_Кафедра комп’ютерних та інформаційних технологій і систем\_

(повна назва кафедри)

**Пояснювальна записка**

**до дипломного проекту (роботи)**

|  |
| --- |
| бакалавра |

(освітньо-кваліфікаційний рівень)

на тему:

|  |
| --- |
| РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІТ- |
| ІНФРАСТРУКТУРИ ПОСТАЧАННЯ БУДІВЕЛЬНИХ МАТЕРІАЛІВ |

Виконав: студент 4 курсу, групи 402-ТК

спеціальності

|  |
| --- |
| 123 Комп’ютерна інженерія |

(шифр і назва напряму)

|  |
| --- |
| Кайда І.С. |

(прізвище та ініціали)

Керівник к.т.н., доцент Скакаліна О.В. В.

(прізвище та ініціали)

Рецензент \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище та ініціали)

Полтава – 2021 року

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**«ПОЛТАВСЬКА ПОЛІТЕХНІКА ІМЕНІ ЮРІЯ КОНДРАТЮКА»**

**НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ТА РОБОТОТЕХНІКИ**

**КАФЕДРА КОМП’ЮТЕРНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ І СИСТЕМ**

**КВАЛІФІКАЦІЙНА РОБОТА БАКАЛАВРА**

**спеціальність 123 «Комп’ютерна інженерія»**

**на тему:**

**«Розроблення програмного забезпечення для ІТ-Інфраструктури постачання будівельних матеріалів»**

**Студента групи 402-ТК Кайди Івана Сергійовича**

Керівник роботи:

кандидат технічних наук,

доцент Скакаліна О.В.

Консультант:

Завідувач кафедри:

кандидат технічних наук,

доцент Головко Г.В.

Полтава – 2021

**РЕФЕРАТ**

Кваліфікаційна робота бакалавра: 108 с., 59 рисунків, 4 додатки, 11 джерел.

**Об’єкт дослідження**: розроблення програмного забезпечення для ІТ-Інфраструктури постачання будівельних матеріалів.

**Мета роботи**: розроблення БД для компанії AmoGusBuilding що займається оптовим постачанням будівельних матеріалів, автоматизація бізнес процесів.

**Методи**: проектування та розробка бази даних з обліку діяльності будівельної компанії з надання послуг по постачанню будівельних матеріалів.

**Ключові слова**: будівельна компанія, будівельні товари, автоматизована інформаційна система, програмне забезпечення, ІТ-інфраструктури.

**ABSTRACT**

Qualifications of the master's robot: 108 p., 59 malunks, 4 supplements, 11 dzherel.

**Object research**: distribution of software security for IT-Infrastructure for the production of alarm materials.

**Meta robots**: Database development for the company AmoGusBuilding is engaged in the wholesale supply of alarm materials, automation of business processes.

**Method**: design and development of the base of data from the scope of the activity of the alarm company with the provision of services for the production of alarm materials.

**Key words**: building company, building comrades, automated information system, security software, IT-infrastructure.

ЗМІСТ

[ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ 6](#_Toc74919435)

[ВСТУП 7](#_Toc74919436)

[РОЗДІЛ 1 ТЕОРЕТИЧНІ ПОНЯТТЯ БАЗ ДАНИХ. ОСНОВНІ ПОНЯТТЯ І ВИЗНАЧЕННЯ 10](#_Toc74919437)

[1.1. Основні поняття баз даних 10](#_Toc74919438)

[1.2. Принципи побудови баз даних та їх життєвий цикл 13](#_Toc74919439)

[1.3. Архітектури баз даних 21](#_Toc74919440)

[1.4. Аналіз сучасних інформаційних систем 23](#_Toc74919441)

[1.5. Впровадження інформаційних систем на підприємствах 30](#_Toc74919442)

[1.6. Види і сучасні системи управління базами даних 34](#_Toc74919443)

[Висновок до розділу 1 43](#_Toc74919444)

[РОЗДІЛ 2 АНАЛІЗ ВИКОРИСТАННЯ АВТОМАТИЗАЦІЇ BUSINESS PROCESS І СУБД ACCESS 45](#_Toc74919445)

[2.1. Опис і системний аналіз предметної області 45](#_Toc74919446)

[2.2. Використання баз даних в будівельних підприємствах 47](#_Toc74919447)

[2.3. Програмний пакет для створення баз даних Microsoft Access 53](#_Toc74919448)

[2.4. Система управління базами даних Access 56](#_Toc74919449)

[Висновок до розділу 2 75](#_Toc74919450)

[РОЗДІЛ 3 ПЕРСПЕКТИВИ ВИБОРУ ТА РОЗРОБКА БД В СЕРЕДОВИЩІ MS ACCESS 76](#_Toc74919451)

[3.1. Обгрунтування вибору СУБД 76](#_Toc74919452)

[3.2. Аналіз практики та застосування СУБД Access для автоматизації бізнес-процесів 77](#_Toc74919453)

[Висновок до розділу 3 92](#_Toc74919454)

[ВИСНОВКИ 93](#_Toc74919455)

[СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ 95](#_Toc74919456)

**ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ І ТЕРМІНІВ**

**ПЗ –** програмне забезпечення.

**БД** – база даних.

**СУБД** – система управління базами даних.

**СКБД** – система контролю базами даних.

**ІС –** інформаційна система.

**ІТ –** інформаційні-технології.

**MS ACCESS –** реляційна система управліня базами даних, розроблена корпорацією Microsoft.

**MS SQL SERVER –** система управління базами данних.

**SQL –** мова програмування, що застосовується для створення та управління даними в реляційній базі даних.

**VBA –** процедурна мову.

**DAO –** Data Access Objects.

**КІС** – корпоративна інформаційна система.

**DB2** – система управління реляційними базами даних.

**ACY –** автоматизовані системи управління.

**ВСТУП**

**Актуальність.** Сучасні системи програмного забезпечення інфраструктури реорганізовуются під європейські стандарти та показують досить непогані успіхи. Нажаль на сьогодні, існує багато технічних аспектів, які потребують покращення. В основному це стосується того, що більшість компаній не володіють достатнім інформаційним потенціалом про власну технічну незабезпеченість, що знижує потенціал розвитку компанії, яка необхідна для покращення власної інфраструктури і ведення бізнес-процесів.

Робітники та керівники в системах ІТ-інфраструктури витрачають багато часу на заповнення паперових документів, адже ведення документообігу дуже часозатратний процес, замість того, щоб автоматизувати і підвищити роботу всієї компанії. Автоматизація є важливим фактором успішності багатьох світових бізнес-гігантів. Адже це не тільки підвищує ресурсоекономність, а також підвищує швидкість виконання задач. В сучасних компаніях та фірмах, покращення власної ІТ-інфраструктури розглядається як основний фактор підвищення успішності у рамках бізнес-розвитку та конкурентноспроможності не тільки в Україні, але й на світовому ринку. Саме тому систематична розробка та апгрейд власного технічного та програмного забезпечення необхідно, але робота із паперами потребує набагато більше часу, людських та матеріальних ресурсів , ніж ведення всього документообігу в базі даних [1, с. 132].

На сьогодні в Україні є велика кількість установ, мета яких – якісно та в повній мірі забезпечити компаніям відповідний рівень технологічної ІТ-інфраструктури.

Українські компанії, як і багато інших використовує програмне забезпечення, зокрема базу даних та СУБД. База даних (БД) – іменована сукупність даних, що відображає стан об’єктів та їх відносин у розглянутій предметній області. Під предметною областю прийнято розуміти деяку область людської діяльності або область реального світу, що підлягають вивченню для організації управління і автоматизації, наприклад, підприємство, внз і т.д. Система контролю базами даних (СКБД) – сукупність мовних та програмних засобів, призначених для створення, наповнення, оновлення та видалення баз даних.

В даний час успішне функціонування різних фірм, організацій та підприємств просто не можливо без розвинутої інформаційної системи, яка дозволяє автоматизувати збір і обробку даних. Зазвичай для зберігання і доступу до даних, що містить відомості про деяку предметну область, створюється база даних.

Програми, за допомогою яких користувачі працюють з базою даних, називаються додатками. У загальному випадку з однією базою даних можуть працювати безліч різних додатків. Наприклад, якщо база даних моделює деякий підприємство, то для роботи з нею може бути створено програму, яка обслуговує підсистему обліку кадрів, інший додаток може бути присвячено роботі підсистеми розрахунку заробітної плати співробітників, третє додаток працює як підсистеми складського обліку, четвертий додаток присвячено плануванню виробничого процесу. При розгляді додатків, що працюють з однією базою даних, передбачається, що вони можуть працювати паралельно і незалежно один від одного, і саме СУБД покликана забезпечити роботу безлічі додатків з єдиною базою даних таким чином, щоб кожне з них виконувалося коректно, то враховувало всі зміни в базі даних, що вносяться іншими додатками [3, с. 79].

Для пошуку інформації в базах даних використовується інформаційно-пошукова система. Інформаційно-пошукова система спирається на базу даних, в якій здійснюється пошук потрібних документів за заявками користувачів.

Впровадження БД надає вагому перевагу, таку як зменшення навантаження на працівників фірми, цілісність документообігу та його автоматизацію, що підштовхує керівників до введення нових методів та технологій для оптимізації роботи бізнес процесу.

**Мета роботи.** Метою даної кваліфікаційної роботи бакалавра є розробка бази даних для компанії AmoGusBuilding. Введення БД в інфраструктуру підприємства допоможе автоматизувати бізнес-процеси з продажу, товарообігу, ведення документації, цілісності даних.

**Напрямки досліджень.** Нами було проведено аналіз ІТ-ринку, і на основі популярних технологій створювали власну БД. Було використано базові технологічні можливості для раціонального використання. Для створення програмного забезпечення використали Microsoft Access – реляційна система управління базами даних, розроблена корпорацією Microsoft, входить в пакет програм Microsoft Office. Завдяки вбудованій мові VBA, в самій Access можна писати програми, що працюють з базами даних. Має вбудовану мову VBA, тому в самій Access можна писати бази даних та програми, що працюють з базами даних.

**Загальні висновки.** Створена БД задовольняє вимоги поставленої задачі. Всі компоненти працюють відповідно завданням. Також дана система має можливість модернізації, переносу на нові технології, платформу:

1. ms sql server;
2. oracle database;
3. sql.

Також при розробці БД було досліджено плюси впровадження даного технічного рішення для підприємств, що ще раз доводить необхідність даного проекту.

**РОЗДІЛ 1  
ТЕОРЕТИЧНІ ПОНЯТТЯ БАЗ ДАНИХ. ОСНОВНІ ПОНЯТТЯ І ВИЗНАЧЕННЯ**

* 1. **Основні поняття баз даних**

В даний час успішне функціонування різних фірм, організацій та підприємств просто не можливо без розвинутої інформаційної системи, яка дозволяє автоматизувати збір і обробку даних. Зазвичай для зберігання і доступу до даних, що містить відомості про деяку предметну область, створюється база даних [73].

База даних (БД) – іменована сукупність даних, що відображає стан об’єктів та їх відносин у розглянутій предметній області. Під предметною областю прийнято розуміти деяку область людської діяльності або область реального світу, що підлягають вивченню для організації управління і автоматизації, наприклад, підприємство.

Система управління базами даних (СКБД) – сукупність мовних та програмних засобів, призначених для створення, наповнення, оновлення та видалення баз даних. Основоположними поняттями в концепції баз даних є узагальнені категорії «дані» і «модель даних». Поняття «дані» в концепції баз даних – це набір конкретних значень, параметрів, що характеризують об’єкт, умова, ситуацію або будь-які інші чинники, Приклади даних: Петров Микола Степанович, $30 і т.д. Дані не володіють певною структурою, дані стають інформацією тоді, коли користувач задає їм певну структуру, тобто усвідомлює їх смисловий зміст. Тому центральним поняттям в області баз даних є поняття моделі. Не існує однозначного визначення цього терміна, у різних авторів ця абстракція визначається з деякими відмінностями але, тим не менш, можна виділити щось загальне в цих визначеннях [68].

Модель даних – це деяка абстракція, яка, будучи застосовні до конкретних даних, дозволяє користувачам і розробникам трактувати їх уже як інформацію, тобто відомості, що містять не тільки дані, але і взаємозв’язок між ними.

За допомогою моделі даних можуть бути представлені об’єкти предметної області та взаємозв’язку між ними. Залежно від виду організації даних розрізняють такі найважливіші моделі БД [60, с. 37]:

1. ієрархічну;
2. мережеву;
3. реляційну;
4. об’єктно-орієнтовану.

В ієрархічній БД дані представляються у вигляді дерева. Подібна структура БД зручна для роботи з даними, упорядкованими ієрархічно. При оперуванні даними зі складними логічними зв’язками ієрархічна модель виявляється занадто громіздкою.

У мережевої БД дані організовуються у вигляді графа. Недоліком мережевої структури є жорсткість структури і складність її організації.

Реляційна БД отримала свою назву від англійського терміна relation (відношення). Була запропонована в 70-му році співробітником фірми IBM Едгаром Коддом. Реляційна БД являє собою сукупність таблиць, пов’язаних відносинами. Перевагами реляційної моделі даних є простота, гнучкість структури. Крім того її зручно реалізовувати на комп’ютері. Більшість сучасних БД для персональних комп’ютерів є реляційними.

Об’єктно-орієнтовані БД об’єднують мережеву і реляційну моделі і використовуються для створення великих БД з даними складної структури.

Бази даних можна розділити на бази даних першого покоління: ієрархічні, мережеві; другого покоління: реляційні; третього покоління: об’єктно-орієнтовані, об’єктно-реляційні.

Програми, за допомогою яких користувачі працюють з базою даних, називаються додатками. У загальному випадку з однією базою даних можуть працювати безліч різних додатків. Наприклад, якщо база даних моделює деякий підприємство, то для роботи з нею може бути створено програму, яка обслуговує підсистему обліку кадрів, інший додаток може бути присвячено роботі підсистеми розрахунку заробітної плати співробітників, третє додаток працює як підсистеми складського обліку, четвертий додаток присвячено плануванню виробничого процесу. При розгляді додатків, що працюють з однією базою даних, передбачається, що вони можуть працювати паралельно і незалежно один від одного, і саме СУБД покликана забезпечити роботу безлічі додатків з єдиною базою даних таким чином, щоб кожне з них виконувалося коректно, то враховувало всі зміни в базі даних, що вносяться іншими додатками [54, с. 101].

Для пошуку інформації в базах даних використовується інформаційно-пошукова система. Інформаційно-пошукова система спирається на базу даних, в якій здійснюється пошук потрібних документів за заявками користувачів.

Розрізняють фактографічні автоматизовані інформаційні системи (АІС), у яких бази даних складаються з форматованих (формалізованих) записів, і документальні АІС, записами яких можуть служити різні неформалізовані документи (статті, листи і т.п.). У фактографічних АІС прикладом форматованих записів можуть служити, скажімо, записи про операції з прийому і видачі грошей в ощадкасі; запис має чотири основних атрибута: дата, характер операції (прийнято, видано), сума, залишок вкладу [30].

Як форматоваі записи може розглядатися кадрова анкета (особовий листок з обліку кадрів). Правда, такі її розділи, як «попередня робота», «поїздки за кордон» та ін. В звичайній анкеті не до кінця формалізовані і мають змінну довжину, тому при автоматизації цього завдання необхідні деякі поправки. Зазвичай буває доцільно фіксувати максимальну кількість позицій в кожному розділі і тим самим вирівнювати довжину записів (у багатьох записів при цьому можуть виникати позиції з порожнім заповненням).

Основним завданням, розв’язуваної в документальних АІС, є пошук документів за їх змістом. Документальна система за завданням користувача видає необхідні йому документи (книги, статті, закони, патенти, звіти і т.д.). У завданні можуть зазначатися відомості про шуканих документах: автор, найменування, час видання, видавництво і т.д. [47, с. 57].

* 1. **Принципи побудови баз даних та їх життєвий цикл**

Класична технологія проектування реляційних баз даних пов’язана з теорією нормалізації, заснованої на аналізі функціональних залежностей між атрибутами відносин. Процес нормалізації має на меті усунення надмірності даних. Нормалізація дозволяє істотно скоротити об’єм інформації і усунути аномалії в організації зберігання даних. Ступінь нормалізації даних може бути різною. Приведення моделі до необхідному рівню нормальної форми є основою побудови реляційної бази даних [37, с. 44].

Нормалізація досягається шляхом перевірки відповідності таблиць ряду умов, визначених у трьох рівнях нормалізації: першої, другої і третьої нормальних формах (існують також і інші рівні).

Перша нормальна форма вимагає, щоб кожне поле таблиці БД було неподільним і не містило повторюваних груп.

Неподільність поля означає, що містяться в ньому значення не повинні ділитися на більш дрібні. Наприклад, якщо в поле «Підрозділ» міститься назва факультету і кафедри, вимога неподільності не дотримується і необхідно виділити назву факультету або кафедри в окреме поле.

Повторюваними є поля, що містять однакові за змістом значення. Наприклад, якщо потрібно отримати статистику продажів чотирьох товарів по місяцях, можна створити поля для зберігання даних про продаж по кожному товару. Однак що робити, якщо товарів не 4, а 104, і як бути, якщо кількість товарів заздалегідь не відомо? Повторювані групи слід усунути, зберігши в таблиці єдине поле «Товар». В результаті отримаємо запис, що містить інформацію про статистику продажів по одному товару, але цей товар може бути будь-яким [22, с. 340].

Друга нормальна форма вимагає, щоб всі поля таблиці залежали від первинного ключа, тобто, щоб первинний ключ однозначно визначав запис і не був надмірний. Якщо ж в будь-якої таблиці є залежність якихось або не ключових полів від частини первинного ключа, слід виділити їх в окрему таблицю, зробивши первинним ключем нової таблиці ту частину первинного ключа, від якої залежать дані поля, і встановити зв’язок «один до багатьох» від нової таблиці до старої.

Третя нормальна форма вимагає, щоб в таблицях небуло транзитивних залежностей між не ключовим полями, тобто щоб значення будь-якого поля, що не входить в первинний ключ, не залежало від значення іншого поля, також не входить у первинний ключ.

Результатом нормалізації є модель даних, яку легко підтримувати, яка не містить невизначеностей в даних і повторень даних.

Після формальних визначень трьох рівнів нормалізації розберемо конкретний приклад і опишемо можливі проблеми. Як приклад буде розглядатися база даних, що містить відомості про відвідувані студентами курси.

Нехай для зберігання цих відомостей використовуються наступні таблиці.

Таблиця 1.1 – Контактні дані студентів що відвідували курси

|  |  |  |
| --- | --- | --- |
| Name (имя) | Phoneno (телефон) | CourseRegistrations (відвідувані курси) |
| Maijorie Green | 415986 | Basic Computing, Database Administration |
| Bun Gringelsby | 707938 | Database Administration, Advanced Hardware Support |
| Anico Yokamoto | 415935 | Advanced Hardware Support |

Таблиця 1.1 містить такі дані студентів як ім’я, номер телефону, курси.

Таблиця 1.2 – Courses (курси)

|  |  |  |
| --- | --- | --- |
| Course (курс) | Lecturer (лектор) | Room (аудиторія) |
| Basic Computing | Meander Smith | 542 South |
| Database Administration | Dean Straight | 221 East |
| Advanced Hardware Support | Dean Straight | 221 East |

В цьому випадку з’являються такі логічні суперечності [33, с. 115]:

1. якщо курс Basic Computing буде закритий, з таблиці буде видалений лектор Meander Smith і аудиторія 542 South;
2. число курсів, на які може записатися студент, обмежена довжиною записи яку допускає поле Course Registrations;
3. важко виконувати пошук значень в поле Course Registrations, а також використовувати його в обчисленнях;
4. в кожної реєстраційної записи повторюється повна назва курсу. В результаті неефективно використовується простір і зростає ймовірність появи неузгоджених даних, якщо назва курсу введено з помилками. Крім того, при зміні назви курсу потрібно проводити пошук і оновлення всіх реєстраційних записів;
5. таблицю Students неможливо індексувати на прізвище, так як в поле name зберігаються повні імена студентів;
6. якщо лектор змінить аудиторію, доведеться оновити відомості про всі викладаються їм курсах.

Проведемо нормалізацію та упорядкування БД в таблицях.

Таблица 1.3 – Впорядковані контактні дані студентів

|  |  |  |  |
| --- | --- | --- | --- |
| StudentsID(код студента) | Firstname (ім’я) | Lastname (прізвище) | Phoneno (телефон) |
| 1001 | Maijorie | Green | 415986 |
| 1002 | Bun | Gringelsby | 707938 |
| 1003 | Anico | Yokamoto | 415935 |

Дані були впорядковані для більш ефективної роботи, кожному студенту було присвоєно індивідуальний номер, імена та прізвища розділено окремо.

Таблица 1.4 – Записи що реєструють діяльнісь кожного студента записи(Registrations)

|  |  |  |
| --- | --- | --- |
| RegID (код запису) | StudentsID(код студента) | Courses (курси) |
| 1 | 1001 | 1 |
| 2 | 1002 | 2 |
| 3 | 1003 | 3 |
| 4 | 1004 | 4 |
| 5 | 1005 | 5 |

Дана таблиця містить дані про реєстраційну діяльність студента, його особистий номерб та код курсів.

Таблица 1.5 – Нормалізація даних про курси та викладачів

|  |  |  |
| --- | --- | --- |
| Course ID(курс) | Course (курс) | LecturerID (код лектора) |
| 1 | Basic Computing | 1 |
| 2 | Database Administration | 2 |
| 3 | Advanced Hardware Support | 3 |

В цій таблиці дані виставлено так, що спершу йдуть номер курсу, його назва і в кінці ID викладача цього курсу.

Таблица 1.6 – Дані викладачів

|  |  |  |  |
| --- | --- | --- | --- |
| LecturerID(код лектора) | Firstname (ім’я) | Lastname (прізвище) | Room (аудиторія) |
| 1 | Meander | Smith | 542 South |
| 2 | Dean | Straight | 221 East |

Між таблицями існують такі зв’язки:

1. Students (студенти) – Courses (курси): відношення «багато до багатьох» через проміжну таблицю Registrations (реєстраційні записи), іншими словами це відношення зведено до двох відносин «один до багатьох»;
2. Students (студенти) – Registrations (реєстраційні записи): відносини «один до багатьох»;
3. Courses (курси) – Registrations (реєстраційні записи): відношення «один до багатьох»;
4. Lecturers (лектори) – Courses (курси): відношення «один до багатьох»;

Очевидні переваги нормалізації цих таблиць:

1. кожна таблиця містить тільки один набір пов’язаних даних. Наприклад, в таблиці Students тепер немає відомостей про відвідуваних курсах;
2. в кожній таблиці є первинний ключ: в таблиці Students – це поле StudentID, в таблиці Registrations – RegID, в таблиці Courses – CourseID і в таблиці Lecturers – LecturerW;
3. відсутні складові поля. Кожне поле описує тільки один атрибут. Наприклад, поле, що містив ім’я і прізвище студента, розбите на окремі поля, які містять ім’я і прізвище студента;
4. відсутні дані, що повторюються. Так, тепер імена лекторів записуються тільки один раз;
5. відсутні поля, що містять кілька значень. Наприклад, кожна реєстраційний запис курсу тепер розташована в окремому рядку таблиці Registrations. Для порівняння погляньте на поле Course Registrations (відвідувані курси) за попередній варіант таблиці Students;
6. кожне поле повністю залежить від первинного ключа. Наприклад, в таблиці Courses немає поля Room. Це пов’язано з тим, що аудиторія залежить не від коду курсу (CourseID), а від коду лектора (LecturerID).

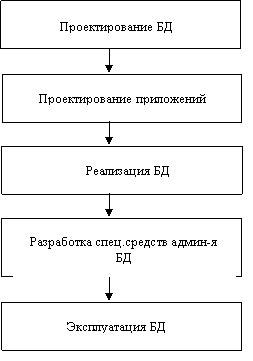
Проект реляційної бази даних – це набір взаємопов’язаних відносин, для яких визначені всі атрибути, задані первинні ключі відносин і задані ще деякі додаткові властивості відносин, які відносяться до принципів підтримки цілісності. Фактично проект бази даних – це фундамент майбутнього програмного комплексу, який буде використовуватися досить довго і багатьма користувачами. Етапи життєвого циклу бази даних (див рис 1.1) аналогічні, в основному, розвитку будь-якої програмної системи, проте в них є певна специфіка, що стосується тільки баз даних [48, с. 92].

Рисунок 1.1 – Етапи життєвого циклу БД

Процес проектування БД являє собою послідовність переходів від неформального словесного опису інформаційної структури предметної області до формалізованого опису об’єктів предметної області в термінах деякої моделі. Можна виділити наступні етапи проектування:

1. Системний аналіз і словесний опис інформаційних об’єктів предметної області.

2. Проектування інфологічної моделі предметної області – частково формалізований опис об’єктів предметної області в термінах деякої семантичної моделі, наприклад, в термінах ЕR-моделі.

3. Логічне проектування БД, тобто опис БД в термінах прийнятої дата логічної моделі даних.

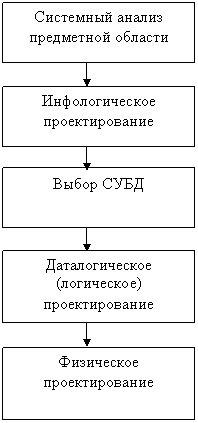
4. Фізичне проектування БД, тобто вибір ефективного розміщення БД на зовнішніх носіях для забезпечення найбільш ефективної роботи програми. Якщо врахувати, що між другим і третім етапами необхідно прийняти рішення, з використанням якої стандартної СУБД буде реалізовуватися наш проект, то умовно процес проектування БД можна уявити послідовністю виконання п’яти відповідних етапів (див. рис. 1.2) [49, с. 105].

Рисунок 1.2 – Етапи проектування БД

Отже дані етапи проектування дають змогу забезпечення найбільш ефективну роботу програми.

* 1. **Архітектури баз даних**

За технологією обробки даних бази даних поділяються на централізовані і розподілені. Централізована база даних зберігається в пам’яті однієї обчислювальної системи. Якщо ця обчислювальна система є компонентом мережі ЕОМ, можливий розподілений доступ до такої бази даних – доступ до неї користувачів різних ЕОМ даної мережі. Такий спосіб використання баз даних часто застосовують в локальних мережах персональних ЕОМ [49, с. 141].

Поява мереж ЕОМ дозволило поряд з централізованими створювати і розподілені бази даних. Розподілена база даних складається з декількох, можливо, що пересікаються або навіть дублюючих один одного частин, які зберігаються в різних ЕОМ обчислювальної мережі. Однак користувач розподіленої бази даних не зобов’язаний знати, яким чином її компоненти розміщені у вузлах мережі, і уявляє собі цю базу даних як єдине ціле. Робота з такою базою даних здійснюється за допомогою системи управління розподіленою базою даних (СУРБД). Дані, що містяться в розподіленої базі даних, їх подання на всіх рівнях архітектури СУРБД і розміщення в мережі описуються в системному довіднику, який сам може бути декомпозирован і розміщений в різних вузлах мережі.

Системи централізованих баз даних з мережевим доступом припускають різні архітектури подібних систем:

1. файл-сервер;
2. клієнт-сервер.

Файл-сервер. Дана архітектура систем БД передбачає виділення однієї з машин мережі як центральної (сервер файлів). На такій машині зберігається спільно використовувана централізована БД. Всі інші машини мережі виконують функції робочих станцій, за допомогою яких підтримується доступ користувальницької системи до централізованої бази даних. Кожен користувач може запускати додаток, розташоване на сервері, при цьому на комп’ютері користувача запускається копія додатку. Файли бази даних відповідно до призначених для користувача запитами передаються на робочі станції, де в основному проводиться обробка. Коли користувач мережі працює з БД, на його комп’ютері з’являється локальна копія загальної БД. Ця копія періодично оновлюється даними, що містяться в БД, розташованої на сервері. Архітектура файл-сервер зазвичай використовується в таких мережах, де є трохи комп’ютерів. Для її реалізації призначені персональні СУБД, наприклад Paradox і DBase. При великій інтенсивності доступу до одних і тих же даних продуктивність інформаційної системи падає [50, с. 94].

Клієнт-сервер. У цій концепції мається на увазі, що крім зберігання централізованої БД сервер бази даних повинен забезпечувати виконання основного обсягу обробки даних. Технологія клієнт-сервер розділяє додаток на дві частини: клієнтську і серверну. Клієнтська забезпечує інтерактивний інтерфейс, сервер забезпечує управління даними, поділ інформації, адміністрування і безпека. Для отримання даних додаток-клієнт формує і відсилає запит віддаленого сервера, на якому розміщена БД. Запит формується на мові SQL, який є стандартом доступу до сервера при використанні реляційних баз даних. Після отримання запиту віддалений сервер направляє його SQL-сервера (сервера баз даних). SQL-сервер – це програма, яка управляє віддаленої БД і забезпечує виконання запиту і видачу клієнтові його результатів – необхідних даних. Вся обробка запиту виконується на віддаленому сервері. Для реалізації архітектури клієнт-сервер зазвичай застосовуються розраховані на багато користувачів СУБД, наприклад Qracle, MS SQL Server, InterBase і ін. Подібні СУБД називають промисловими, так як вони дозволяють організувати інформаційну систему, що складається з великого числа користувачів [82].

* 1. **Аналіз сучасних інформаційних систем**

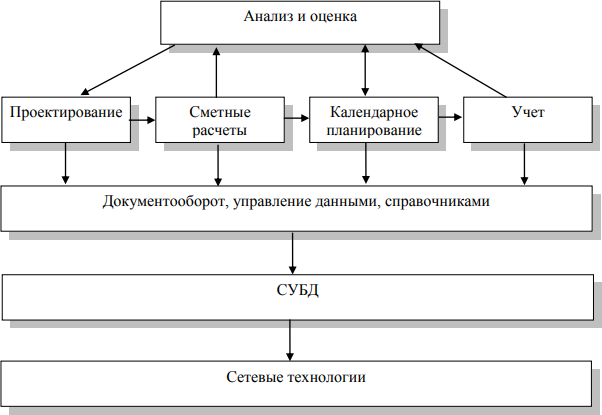
На сьогоднішній день на Українському ринку запропоновано безліч систем як вітчизняних так і зарубіжних розробників.

Дані програмні продукти відрізняються не тільки за вартістю, але і способами реалізації одних і тих же функціональних вимог, платформами розробки і технологіями. Серед даного різноманіття є свої лідери, проте це не заважає появі на ринку нових рішень.

Необхідно провести аналіз сильних і слабких сторін існуючих готових систем, оптимальних для даного підприємства, вірно оцінити вартість і витрачені сили.

В результаті аналізу сформується висновок про необхідність придбання готового продукту або ж розробки власного.

З урахуванням числа будівельних організацій, чисельності їх працівників та ролі будівельного комплексу в економіці України завдання управління будівельними організаціями є одними з найбільш актуальних, а інформаційні системи будівельних організацій – одними з найбільш затребуваних. У дисертації розглянуті особливості інформаційних систем будівельних організацій, їх ключові компоненти, основні тенденції їх розвитку. На рисунку 1.3 наведена модель структури інформаційної системи будівельної організації [8, с. 103].

Рисунок 1.3 – Детальна структура ІС будівельної організації

Аналіз наукових і технічних публікацій і практики використання інформаційних систем в діяльності будівельних організації дозволяє виділити наступні особливості автоматизації будівельних організацій: – орієнтація на проект. В силу специфіки будівництва проект виявляється ключовим елементом системи. Методи, технології та програмні засоби управління проектами відіграють значну роль в інформаційній системі будівельної організації. При цьому різні підсистеми розглядають проект з різних точок зору і з різним ступенем деталізації. Високий рівень ризику для організації. У багатьох випадках будівельна організація бере на себе відповідальність за ту чи іншу сторону будівництва великого об’єкта. Для багатьох таких організацій успішне виконання робіт по одному проекту визначають успішність функціонування компанії в цілому. Проектування Кошторисні розрахунки Аналіз і оцінка Документообіг, управління даними, довідниками Календарне планування СУБД Облік Мережеві технології взаємне переплетення технічної і економічної інформації. Наприклад, кошторис є невід’ємною частиною проектно-кошторисної будівельної документації та сама залежить від параметрів проекту орієнтація на холдинги. У багатьох випадках будівельні організації існують в рамках великих будівельних холдингів. Такий холдинг може включати кілька організацій, які здійснюють проведення будівельно-монтажних робіт, а також кілька організацій виконують вузькоспеціалізовані завдання, наприклад, перевезення будівельних вантажів, матеріально-технічне постачання, проектування. Таким чином, інформаційні системи різних організацій повинні інтенсивно взаємодіяти між собою. Інтеграція різнорідних систем. Інформаційні системи будівельних організацій включають безліч різних систем: САПР, системи управління проектами, системи кошторисних розрахунків, системи фінансового аналізу, облікові системи. Всі ці компоненти повинні взаємодіяти між собою. Високий рівень відповідальності. Результати функціонування будівельних інформаційних систем втілюються в кінцевому підсумку у вигляді виконуваних робіт і зведених будинків. Відповідно набагато вище в порівнянні з іншими галузями і вимоги до безпеки: до безпеки для життя і здоров’я людей, безпеки для матеріальних об’єктів, безпеки з точки зору охорони навколишнього середовища. Інформаційна система повинна сприяти дотриманню державних будівельних норм і правил. Значний обсяг державного регулювання. По-перше, в будівельній галузі від підприємств потрібно виконання численних норм і правил, державних нормативів і розцінок, які інформаційна система повинна, таким чином, підтримувати. По-друге, твердження державними органами значної частки проектних рішень. По-третє, представники державних органів мають право і обов’язок контролювати дотримання норм і правил безпосередньо під час виконання будівництв і по його закінченню. Це також вимагає додаткових вимог до інформаційної системи, яка повинна підтримувати державні затверджені форми документів, видавати необхідні звіти і оперативну інформацію. Особливий характер взаємин між учасниками будівельного процесу. Специфіка будівельного комплексу полягає і в тому, що учасники будівельного процесу (інвестор, замовник, проектувальник, генеральний підрядник, субпідрядники) інтенсивно взаємодіють між собою на всьому протязі будівництва. При цьому виникає значний обсяг інформаційного обміну. Інформаційні системи систем різних суб’єктів будівельного процесу, таким чином, повинні бути пов’язані між собою і підтримувати можливість обміну електронними документами. Перераховані особливості дають уявлення про специфіку інформаційних будівельних організацій, пояснюють існуюче велике число програмних систем для будівництва, і кажуть про актуальність дослідження споживчої якості інформаційних систем в будівельній галузі [5, с. 209].

Сьогодні розвиток інформаційних систем будівельних організацій характеризується наступними основними тенденціями:

1. широке використання мережевих технологій для обміну даних між підсистемами, а також інформаційними системами різних організацій;
2. наближення інформаційних систем безпосередньо до місць виконання робіт. З використанням мобільних і інтернет-технологій фахівець може, перебуваючи на об’єкті будівництва, знайти необхідне рішення за допомогою інформаційної системи, а виконроб може дати звіт про хід виконання робіт;
3. інтеграція інформаційних систем організацій на основі галузевих стандартів;
4. вдосконалення методів управління на основі використання ІС. Перш за все, організації перебудовуються на проектне управління, яке найбільш повно відповідає специфіці будівельного виробництва. Зсув від облікових завдань до підтримки прийняття рішень;
5. можливості більшості використовуваних сьогодні інформаційних систем відносяться до обліку виконуваних робіт і складання звітності для державних органів, а також до виконання бухгалтерських завдань, таких як визначення вартості будівництва та нарахування заробітної плати. У той же час, в будівництві намічається тенденція використання інформаційних систем в першу чергу для підтримки прийняття рішень керівниками компанії і технічними фахівцями.

На рисунку 1.4 представлена виділена сукупність найбільш значущих ознак класифікації компонентів ІС будівельних організацій. На малюнку також представлені можливі варіанти систем для кожного з класифікаційних ознак. Така класифікація дозволяє систематизувати знання про види інформаційних систем в будівництві та бути корисна при їх виборі, проектуванні, розробці, впровадженні. Наприклад, ознака «Ланка договірних відносин в будівництві» характеризує, для якої організації, з беруть участь в будівельному процесі, призначена система. Залежно від ролі організації та функції системи будуть різні. У таблиці представлені можливі варіанти і приклади класифікаційної ознаки [26].

Рисунок 1.4 – Признаки классифікації програмного забезпечення для будівених організацій

Специфіка будівельного бізнесу і особливості використання інформаційних систем будівельними організаціями визначає склад показників споживчої якості для таких систем, а також їх окремих компонентів.

Таблиця 1.7 – Варіанти інформаційних систем для класифікаційної ознаки «Ланка договірних відносин» 1

|  |  |
| --- | --- |
| Варіант | Приклад |
| Інвестор | Project Expert |
| Замовник або забудовник | ГЕКТОР: Календарний план будівництво объектів ІС:Замовник будівництва 1.0 |
| Підрядник | ІС:Підрядник будівнцтва 4.0  Держбудсмета 2.0 Будівник  Менеджер будівництва 2.0 |
| Проектувальник | Адепт: Проект ArchiCAD ПИР  Гектор:Пректувальник-будівник |
| Державні установи | Галузева автоматизована система інформаційного забезпечення Укрархітектури |
| Універсальна система | Smeta.ru A0 |

Ознаки класифікації програмного забезпечення в будівельному комплексі: за масштабом, за рівнем спеціалізації, за функціональним призначенням, за ступенем інтеграції, за рівнем використання мережевих технологій, по ланці, по стадії для малих підприємств, для середніх підприємств, для великих підприємств, для холдингів, універсальні спеціалізовані (будівництво) адаптовані, вузькоспеціалізовані, облікові системи, кошторисна програмне забезпечення системи автоматизованого проектування системи, календарного планування та управління проектами, індивідуальні системи, мережеві системи, онлайнові системи, мобільні системи, незалежні системи надбудови над універсальними системами, інтегровані програмні комплекси, унітарні програмні комплекси проектування підготовки будівництва управління будівництвом здачі / приймання, експлуатації, інвестор, замовник, генпідрядник, підрядник, проектувальник, державні органи. Аналіз науково-технічної та спеціальної літератури, технічної документації, матеріалів мережі Інтернет дозволив виділити ряд основних характеристик споживчої якості для ІС будівельних організацій. Для порівняльної оцінки характеристик споживчої якості ІС будівельних організацій був використаний методом аналізу ієрархій Т. Сааті. Метод передбачає побудову дерева цілей і альтернатив, порівняння і ранжування. Основна ідея застосування такого підходу до проблеми оцінки споживчої якості ІС будівельних організацій полягає в тому, що цінність кожного з показників споживчої якості визначаються тим, якою мірою він сприяє досягненню цілей організації. Для виділення вузлів дерева і парних порівнянь залучалися експерти-фахівці з досвідом використання інформаційних систем в будівельному бізнесі. Побудоване дерево цілей представлено на рисунку 1.5 [28, с. 117].



Рисунок 1.5 – Ієрархія цілей використання и характеристик споживчих якостей інформаційної системи будівельних організацій

На верхньому рівні встановлена головна мета інформаційної системи – забезпечення досягнення основних цілей будівельної організації. У свою чергу, головна мета (нульовий рівень) складається з двох підцілей (цілей першого рівня): забезпечення безпеки з точки зору життя, здоров’я людей, а також екологічної безпеки; отримання прибутку. Досягнення основних цілей будівельної організації, забезпечення безпеки, дотримання норм і правил, безпомилковість розрахунків, підтримка прийняття управлінських рішень, зниження витрат праці на управління і облік, прискорення і вдосконалення бізнес-процесів. Для досягнення цих підцілей виділяються цілі другого рівня дотримання норм і правил, безпомилковість розрахунків і проектних рішень (забезпечують досягнення мети «Безпека»); підтримка прийняття управлінських рішень, забезпечення економії за рахунок зниження витрат праці на управління, підтримка проектно-кошторисної діяльності, реінжиніринг бізнес-процесів (забезпечують досягнення мети «Отримання прибутку»). Цілі другого і третього рівня зв’язуються між собою. Всі цілі третього рівня пов’язані з усіма характеристиками споживчої якості [29].

* 1. **Впровадження інформаційних систем на підприємствах**

Ефективність технологій підштовхує керівників прийняти рішення про впровадження системи в діяльність підприємства. Але перед тим як купувати ліцензії і наймати відому компанію для впровадження, необхідно отримати основні поняття про цей процес, щоб контролювати хід впровадження на власному підприємстві.

Після сформувалася думки про те, що компанії необхідно впровадження інформаційної системи управління підприємством, потрібно визначитися, хто буде займатися проектуванням, розробкою та впровадженням інформаційної системи. Існує кілька підходів до запуску проектів такого роду на підприємстві. Розглянемо їх докладніше [21, с. 303].

Можливо укласти контракт з великою компанією, яка впроваджує ІС. До переваг даного підходу можна віднести досвід компанії-аутсорсера і окремих її фахівців, а також наявність власних проектних напрацювань. До недоліків відносять вартість робіт, можливу плинність кадрів і можливість того, що за гучним ім’ям можуть стояти не найкращі фахівці [21, с. 315].

Також можна запросити невелику, регіональну IT-компанію. Однозначним плюсом є висока ймовірність того, що впровадження автоматизованої інформаційної системи стане пріоритетним проектом для неї. Якщо проект має бути великий, а значить довгий, варто побоюватися раптових змін керівництва, фахівців і пріоритетів невеликих фірм впроваджувачів [6].

Впровадження силами власного IT-відділу привертає відсутність додаткових витрат, постійний зв’язок з фахівцями та можливість особисто керувати проектом. Однак тут криється і велика небезпека – фахівці IT- відділу, часто залежать від користувачів і керівництва, повністю орієнтуються їх вирішення, в тому числі не завжди правильні [5, с. 219].

Крім того, можна запросити експерта. Відмінний спосіб заощадити і отримати фахівця в потрібній області. До недоліків можна віднести необхідність високої організованості всіх співробітників компанії, залежність успіху від однієї людини і формальну відповідальність за проект.

Практика показує, що управління впровадженням інформаційних систем краще довірити досвідченим фахівцям.

Перейдемо до розгляду етапів впровадження інформаційної системи. Успішні впровадження інформаційних систем включають в себе досить багато важливих і корисних для підприємства етапів. Крім безпосередньо початку роботи в ІС, вони допомагають компанії впорядкувати ключові бізнес-процеси і виявити проблемні місця. Чим більша компанія, тим більше прийнято співробітників, чиї функції і кваліфікація викликають питання. Грамотне впровадження інформаційної системи виявить їх [24].

Існує 7 основних етапів впровадження інформаційної системи. Розберемо кожен етап докладніше.

В першу чергу необхідно визначити мету впровадження. Багато керівників вищої ланки поверхово ставляться до цього етапу, але насправді він задає напрямок всьому впровадження ІС.

Потім необхідно обстеження бізнес-процесів компанії. В цей етап входять інтерв’ю з менеджментом, рядові працівники приймають, складання схем по кожному процесу. На виході виходить уточнення цілей впровадження і можливість попередньо оцінити обсяг робіт і вартість.

Далі йде складання проекту, технічного завдання і регламенту. У цих документах повинні бути описані всі бізнес-процеси, які беруть участь у впровадженні ІС. Важливо складати проект впровадження максимально докладно, із зазначенням необхідних даних, їх структури, алгоритмів дій, робочих місць [19, с. 122].

Після цього необхідно підготувати фахівців. Співробітники компанії при початку впровадження повинні знати, що від них вимагається, щоб не затримувати виконання роботи. Також адміністратори і розробники компанії повинні почати розбиратися в інформаційній системі. Тобто співробітники розширюють свої знання на благо компанії.

Важливий етап – налаштування інформаційної системи відповідно до специфіки підприємства. В цей етап включається: розмежування прав на функціонал системи для співробітників; початкове заповнення даних; настройка алгоритмів розрахунків, створення необхідних звітів.

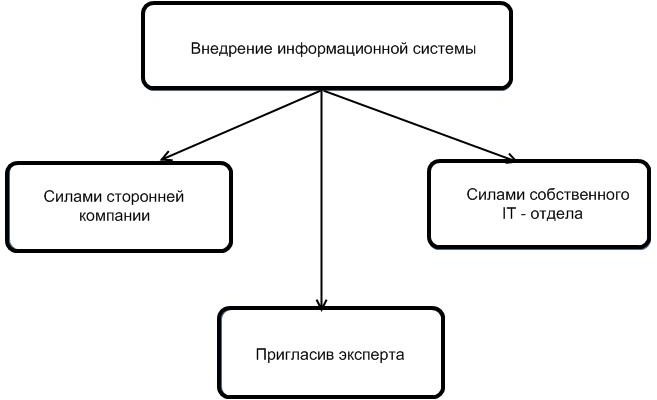
Далі тестування інформаційної системи. На цьому етапі можуть виявитися проблеми впровадження в розрізі алгоритмів або необхідність в нових звітах.

Потім слід дослідна експлуатація з реальними даними. Найчастіше на цьому етапі багато співробітників компанії виконують більше роботи. Їм доводиться не тільки працювати, як раніше, а й відображати свої дії вінформаційній системі. Потрібна максимальна дисципліна і зосередження зусиль всіх учасників впровадження. Кінцевим результатом має стати збіг даних інформаційної системи з реальним станом справ [39, с. 108].

Промислова експлуатація, де здійснюється перехід співробітників на повноцінну роботу в інформаційній системі. Повинна бути організована технічна підтримка користувачів.

І нарешті, завершення проекту. Основним результатом етапу є підписані посадові інструкції, розмежування обов’язків підрозділів та їх взаємодії. Корпоративна інформаційна система запущена на підприємстві.

Отриманий матеріал, аналіз якого дозволив зробити висновок, що існує кілька підходів до впровадження інформаційних систем на підприємстві. Дані підходи показані на рисунку 1.6.

Рисунок 1.6 – Підходи впровадження інформаційних систем

З вищесказаного випливає, що при впровадженні інформаційних систем необхідно пройти всі етапи. Рисунок 1.7 ілюструє послідовність етапів впровадження ІС.

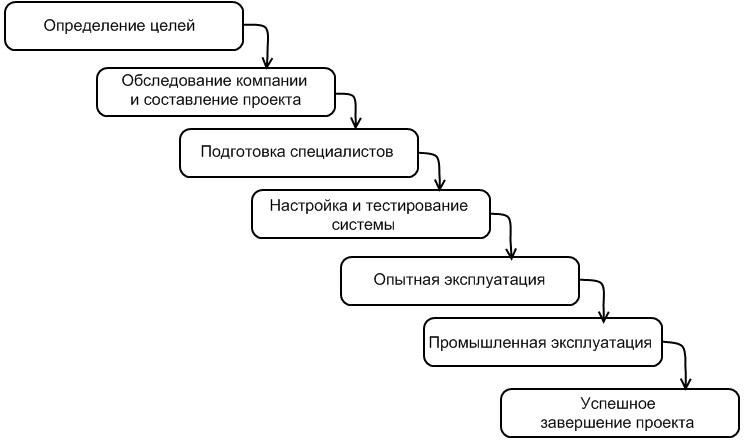


Рисунок 1.7 – Етапи впровадження інформаційних систем

Розвиток сучасних технологій ведуть до зростання складності впроваджуваних інформаційних систем, які вбирають в себе як технічні новинки, так і нові віяння і рішення в сфері ведення бізнесу. З упевненістю можна сказати, що на сьогоднішній день наявність КІС на підприємстві просто необхідно, інакше організація ризикує втратити конкурентоспроможність і просто зникнути. А значить, розвиток методологій впровадження інформаційних систем буде розвиватися в найближчому майбутньому бурхливими темпу [41].

* 1. **Види і сучасні системи управління базами даних**

В даний час установка і адміністрування баз даних – це набагато менш складний процес, ніж ще деякий час назад. Проектування і управління базою даних здебільшого автоматизовані. Програмне забезпечення, яке дозволяє вирішити цю задачу – створювати базу даних, оновлювати зберігається в ній інформацію – і забезпечує швидкий і зручний доступ до неї з метою перегляду та пошуку інформації, називається системою управління базами даних.

Система управління базами даних створює на екрані комп’ютера певне середовище для роботи користувача (призначений для користувача інтерфейс). Крім того, СУБД має певні режими роботи і систему команд. На основі СУБД створюються і функціонують ІС. Системи управління базами даних грають виняткову роль в організації сучасних будівельних, інструментальних і дослідницьких ІС [40].

Основними режимами роботи з СУБД є створення, редагування, управління і пошук в базі даних. Для роботи в кожному режимі існує своя система команд СУБД. Будь-яка робота користувача з базою даних будується в формі алгоритму, який складено з цих команд. Такі алгоритми можуть виконуватися в режимі прямого виконання (віддана команда відразу виконується) і в режимі автоматичного виконання, в програмному режимі [42].

На сьогоднішній день існує досить багато СУБД різного призначення. У сучасних ІС стандартом є реляційні СУБД, дані в яких зберігаються в таблицях. Всі ці системи мають свої відмінності. Перерахуємо ключові параметри, які є важливими як для користувачів, так і для розробників системи.

Спосіб доступу до бази даних:

1. файл-серверні СУБД;
2. клієнт-серверні СУБД;
3. вбудовувані СУБД.

У файл-серверних СУБД (MicrosoftAccess, Paradox, FoxPro, dBase та інші) Для додатків відкритий загальний доступ до всіх файлів бази даних, що зберігаються в якомусь разделяемом файловому сховищі, вони можуть спільно обробляти ці дані. Кожна програма обробляє дані самостійно. Але дана технологія на сьогодні вважається застарілою, тому що у файл-серверних СУБД відсутні багато переваг клієнт-серверних, таких як: паралелізм запитів, кешування даних, висока продуктивність, і, в той же час, їм притаманний ряд недоліків, що в свою чергу призводить до зниження продуктивності і надійності.

У клієнт-серверних СУБД (InterBase, MicrosoftSQLServer, Firebird, Oracle, MySQL, PostgreSQLu ін.) Вся обробка даних здійснюється на сервері, де зазвичай зберігаються дані, але доступ до файлів даних має тільки один сервер, одна система – сама система управління базами даних. При цьому додатки-клієнти лише посилають запити до СУБД на обробку та отримання даних і отримують відповіді; безпосереднього доступу до файлів даних програми-клієнти не мають [48, с. 286].

Вбудовувані СУБД (MicrosoftSQLServerCompact, FirebirdEmbedded, SQLite, і ін.) Поставляються в складі готового інформаційного продукту, для них потребує окремого встановлення. Завданням СУБД цього типу є локальне зберігання даних програми. На колективне використання в мережі вони не розраховані [50, с. 83].

За схемою ліцензування СУБД бувають:

1. безкоштовні;
2. комерційні промислові (в більшості випадків з наявністю безкоштовної обмеженою версії).

Практично всі файл-серверні і вбудовуються СУБД є безкоштовними.

Багато виробників промислових СУБД надають можливість користуватися безкоштовними версіями своїх продуктів, що мають обмеження по продуктивності і по функціоналу на відміну від повнофункціональної версії СУБД.

До плюсів вільних СУБД можна віднести:

1. безкоштовні;
2. не вимогливі до ресурсів;
3. широкі функціонал і продуктивність; надійність.

Мінуси:

1. немає гарантій, що через певний час проект не припинить свою роботу і підтримку продукту;
2. складно знайти освіченого фахівця для роботи з СУБД типу PostgreSQL або Firebird.

Плюси комерційних СУБД:

1. висока продуктивність;
2. надійність;
3. масштабність;
4. підтримка розробником;
5. документація;
6. вбудовані інструменти для адміністрування і розробки.

Такі компанії як, Microsoft, IBM або Oracle, не припинять підтримку своїх систем.

Мінуси комерційних СУБД:

1. висока вартість;
2. висока вимогливість до ресурсів.

В основі класифікації СУБД лежить використовувана модель баз даних, що дозволяє виділити кілька класів СУБД: мережеві, ієрархічні, реляційні, об’єктні та ін. Найпоширенішими СУБД на даний момент є реляційні.

У реляционном підході організації СУБД передбачається наявність набору відносин, які пов’язані між собою. Зв’язок в даному випадку це асоціювання двох або більше відносин. База даних, що володіє обмеженою структурою і не має зв’язків між таблицями – не може називатися реляційної. Запити до таких баз даних повертають таблицю, яка може брати участь у наступному запиті повторно. Дані в одних таблицях пов’язані з даними інших таблиць, звідси і пішла назва «реляційні» [51, с. 122].

Реляційний підхід в побудові СУБД має ряд переваг: невеликий набір абстракцій, що дозволяє легко моделювати більшу частину поширених предметних областей і допускає точні формальні визначення, залишаючись зрозумілими; наявність простого і в той же час потужного математичного апарату, який спирається головним чином на теорію множин і математичну логіку і забезпечує теоретичний базис реляційного підходу до організації баз даних; можливість ненавігаційній маніпулювання даними без необхідності знання конкретної фізичної організації баз даних у зовнішній пам’яті [52, с. 84].

Реляційна модель має суворе обгрунтування з точки зору теорії. Для визначення і керування базами даних був створений декларативний мову SQL. Сильні сторони реляційної моделі – простота, придатність для систем інтерактивної обробки транзакцій, забезпечення незалежності від даних. Однак реляційна модель даних і реляційна СУБД, зокрема, мають і деякі мінуси:

1. ограниченность использования в областях, где достаточно сложные структуры данных;
2. специфіка реалізації реляційної моделі не дає адекватно відображати реальні зв’язки між об’єктами в описуваної предметної області.

ORACLE, Informix, IBM (DB2), Sybase, Microsoй, Progress та інші це значущі в даний час фірми виробники реляційних СУБД. Програми виробників СУБД розраховані на роботу на різних типах комп’ютерів і на різних операційних системах. Також виробники СУБД звернули увагу на програми, що працюють на настільних комп’ютерах, такі як dBase, FoxPro, Access. Дані СУБД призначені для роботи на PC і вирішують локальні завдання на одному PC або невеликій групі. Часто дані СУБД використовуються, як дзеркальне відображення невеликої частини загальної корпоративної СУБД, для мінімізації необхідних апаратних і ресурсних витрат для вирішення невеликих завдань [55, с. 89].

Різні СУБД працюють під управлінням різних операційних систем і апаратної частини. UNIX, VAX, Solaris, Windows – ці популярні операційні системи. Залежно від обсягу храненімих даних, кількості користувачів, які здійснюють одночасний доступ до даних, складності завдань – використовуються різні СУБД на різних платформах.

Виникнення об’єктних баз даних було визначено необхідністю вирішувати завдання, які пов’язані з обробкою і зберіганням складних багатозв’язних даних, а також погано структурованою і неструктурованою інформацією: текстом, зображеннями, музикою і т.д. Об’єктна СУБД найкраще підходить для опису такого роду даних, на відміну від реляційних СУБД, де додавання нового типу даних досягається ціною втрати продуктивності або за рахунок різкого збільшення термінів і вартості розробки додатків.

Об’єкти, на відміну від реляційних таблиць, тісно пов’язують дані і програмний код. Об’єкт являє собою пакет, який містить значення всіх даних цього об’єкта і копію всіх його кодів.

Методи об’єкта направляють повідомлення для взаємодії з іншими методами цього ж або інших об’єктів.

Об’єктні СУБД реалізують величезний набір функцій, який характерний системам управління базами даних, плюс можливості об’єктного програмування. Виходить, що ми отримуємо всі переваги СУБД поряд з потужним об’єктним мовою програмування (серед них С ++, Java) об’єктів бази [56, с. 97].

У об’єктної технології складність структур даних знаходяться всередині самих об’єктів, а доступ до інформації здійснюється через простий інтерфейс.

Основними поняттями, з якими працює ця модель, є наступні:

1. спадкування – породжує один клас об’єктів з іншого. Новий клас зберігає всі властивості і методи свого «батька», може мати додаткові властивості і методи, характерні тільки для нього;
2. інкапсуляція – дає можливість описувати об’єкт незалежно від рівня складності або класу, може мати певне число загальнодоступних властивостей і методів. Додатком не обов’язково знати, як об’єкт влаштований і діє зсередини. Воно взаємодіє тільки з властивостями і методами об’єкта;
3. поліморфізм – значить, що методи, що належать різним класам, можуть використовувати один і той же інтерфейс незалежно від конкретної реалізації цих методів.

Існує ще дві особливості об’єктного підходу – типізація і збережених. Типізація захищає розробника від некоректного використання в прикладних програмах об’єктів одного класу замість іншого. Збережених дозволяє об’єкту продовжувати існувати в системі після завершення процесу, це є найважливішим для концепції об’єктних СУБД [58, с. 100].

Об’єктні СУБД забезпечують инкапсуляцию логіки і даних в одному об’єкті; підтримують складні типи даних і роботу на більш високому рівні абстракції, що дозволяє з одного боку створювати складні структури даних, в тому числі мультимедійні, а з іншого забезпечити простоту їх розвитку.

Популярність об’єктних СУБД в наш час сильно зростає, що пов’язано з широким спектром можливостей щодо їх застосування для побудови ІС корпоративного рівня.

Ще одна класифікація СУБД заснована на методах організації зберігання і обробки даних, тому їх поділяють на централізовані та розподілені. Централізована СУБД зберігається на одному комп’ютері, але користувач працює з базою даних через віддалений доступ.

Розподілена СУБД – перелік програм, який призначений для управління розподіленою базою даних і дозволяє зробити розподіл інформації «прозорою» для кінцевого користувача.

Термін «прозорість» означає, що для кінцевого користувача повинен бути повністю прихований той факт, що розподілена база даних складається з декількох фрагментів, які можуть бути розміщені на декількох комп’ютерах, розташованих в мережі і до неї можливий паралельний доступ декількох користувачів [61, с. 201].

Основний напрямок розподіленої СУБД полягає в забезпеченні засобів інтеграції локальних баз даних, розміщених в деяких вузлах комп’ютерної мережі, для того, щоб користувач, що працює в будь-якому вузлі мережі, мав доступ до всіх цих баз даних як до однієї.

PostgreSQL це високопродуктивна об’єктно-реляційна СУБД з відкритими вихідними текстами. Вона може відкриватися на всіх основних платформах, включаючи Linux, Windows, MacOSX. Вона має повну відповідність з ACID, повністю забезпечує підтримку ключів, об’єднань, уявлень, тригерів, і збережених процедур. У PostgreSQL можуть виконуватися збережені процедури, написані на різних мовах програмування, включаючи C / C ++, Java, Perl, Ruby і власному PL / pgSQL, який схожий з Oracle’sPL / SQL. У стандартній бібліотеці функцій є безліч вбудованих функцій – від базових строкових і математичних операцій до криптографічних функцій і функцій, які забезпечують сумісність з Oracle. Також в PostgreSQL є засоби розробки, що дозволяють створювати призначені для користувача типи даних разом з операторами і функціями, що описують їх поведінку [77].

Поряд з мовами, які можуть бути задіяні при написанні збережених процедур, є і велика кількість інтерфейсних бібліотек, що дозволяють як компільовані, так і інтерпретується мов взаємодіяти з PostgreSQL. Це інтерфейси для ODBC, Perl, Java (JDBC), С, С ++, Lisp, Scheme, PHP, Qt та інші.

Також, вихідний код PostgreSQL доступний під ліцензією BSD, найліберальнішої з відкритих ліцензій. Ця ліцензія дає право на вільне використання, модифікацію і поширення PostgreSQL в будь-якій формі, з закритим або відкритим вихідним кодом. Таким чином, PostgreSQL є не тільки потужною системою управління базами даних, яка дозволяє організовувати діяльність фірми, а й майданчиком розробки для створення додатків, які вимагають використання реляційної СУБД [77].

DB2 представляє собою сімейство систем керування базами даних, які віпускає корпорація IBM. У IBMDB2 вікористовується загальноприйнята на клієнт-серверних архітектура реляційної СУБД, з функцією забезпечення зберігання інформації на сервері і можливістю підключення додатків КЛІЄНТІВ до баз даних через мережу або локально.

Також в DB2 здійсненій механізм підтримки всіх поширених промислових стандартів доступу додатків до даних, включаючи інтерфейси JDBC и ODBC, стандартна мова Запитів SQL, роботу з типами текстових табличних форм.

Характерними особливостями DB2 є:

1. Засоби контролю і розмежування доступу: масштабність, обмежена тільки наявними обчислювальними потужностями, і забезпечує максимально економічне і ефективне використання обчислювальних ресурсів.
2. Наявність розвиненої інтегрованої системи резервного копіювання та відновлення даних.
3. Повний набір технологій для побудови «класичних» аналітичних сховищ даних.
4. Висока відмовостійкість.

Microsoft SQL Server являє собою систему керування базами даних. Microsoft SQL Server розроблена компанією Microsoft. В якості мови запитів використовується мова стандарту ANSI / ISOTransact-SQL з розширеннями. Система працює з базами даних великих і середніх розмірів [73].

SQL Server є базою даних, наповненою інформацією, яка може розширюватися без втрати швидкості операцій із записами в розрахованому на багато користувачів режимі. Нові користувачі можуть додаватися шляхом удосконалення обладнання.

Завдяки тому, що мережева безпека пов’язана безпосередньо з сервером безпеки, це дозволяє максимально захистити дані від несанкціонованого доступу. Оскільки функції безпеки реалізовані на рівні користувача, їх доступ до запису даних може бути обмеженим, що допомагає захистити їх тим самим від пошуку або модифікації, вказавши доступ на рівні користувача привілеєм. До того ж, з даними, які зберігаються на окремому сервері, сервер працює як шлюз, захищаючи від несанкціонованого доступу [53, с. 75].

SQL Server виконує обробку запитів від користувачів і відправляє їм тільки результати. Виходить, що через мережу передається мінімальна інформація, це прискорює час відповіді і дозволяє усувати вузькі місця в мережі. Це також, робить SQL Server оптимальної базою даних для використання в інтернеті.

Технічне обслуговування SQL Server зовсім просте і не вимагає великих знань і умінь. Можливі зміни в структурі даних, а також резервне копіювання під час роботи, без зупинки сервера.

Два основних мови розробки додатків використовується для добування інформації з даних SQL Server – це С ++ і VisualBasic. Ці мови – частина пакету VisualStudioMicrosoft.

Ця система управління базами даних, перш за все, вирізняється високою надійністю. Це можна здійснити за рахунок застосування різних базових технологій, таких як створення відмовостійких кластерів, надання різноманітних засобів для роботи з журналами.

**Висновок до розділу 1**

В результаті роботи над матеріалом було виявлено, що інноваційна діяльність є важливим фактором у ІТ-Інфраструктурі підприємства. Нововведення впливають на конкурентоспроможність та позиції фірми у сфері її функціонування. Кожному підприємству слід займатися інноваційною діяльністю коли їх потребує підприємство. Для визначення доцільності інновацій виробник має проводити аналіз ринку технологій, який вкаже, що більш вигідно – залишитися та впроваджувати нові технології на підприємства.

Важливе місце у діяльності фірми посідає інноваційний процес, який характеризується створенням, контролем та впровадженням нових технологій в свою структуру. Цей процес є безперервним і діє після впровадженням, адже нововведення має здатність удосконалюватися та набувати нових властивостей.

Інноваційна діяльність у сфері ІТ ніколи не стояла на місці. Адже всі світові бізнес-гіганти протягом свого існування займалася необхідними новаціями і завдяки цьому стала лідером в своїй сфері.

**РОЗДІЛ 2  
АНАЛІЗ ВИКОРИСТАННЯ АВТОМАТИЗАЦІЇ BUSINESS PROCESS І СУБД ACCESS**

* 1. **Опис і системний аналіз предметної області**

Розробимо базу даних, яка буде зберігати в собі інформацію про діяльність компанії будівельних матеріалів, постачання, постачальників і товар, вводити нові і виводити звіти: про персонал і їх окладах, фірмах і товарах, постачання і постачальників, а також про складах.

Компанія носить назву AmoGusBuilding. До нього прикріплений склад, на якому зберігаєтся товар.У кожного товару є свій код. Клієнти замовляють товар з складу, звідки потім його доставляють замовникам. Необхідно вести облік того, який товар ппродаєтся, в якій кількості і за якою ціною. А так же через який час його реалізували, скільки і якому клієнту – можна визначити за контактними даними замоника та накладними. Крім того, після поставки товару, необхідно оновлювати записи про його кількость і, можливо, ціною. І, відповідно, повинен віддалятися реалізований товар.

Встановимо атрибути, які повинні міститися в базі даних. Для кожного атрибута повинно бути визначено безліч можливих значень. Перерахуємо імена атрибутів і їх короткі характеристики:

**Адреса** – місце знаходження клієнта або замовника (приклад можливого значення: м. Кременчуг, вул.Пролетарська, 57).

**Телефон** – телефон клієнта, постачальника (можливе значення: 380 76-61-85).

**Прізвище** – прізвище клієнта магазину.

**Ім’я** – ім’я клієнта магазину.

**Код товару** – індивідуальний код товару (приклад можливого значення: 325).

**Найменування товару** – найменування товару (приклад можливих значень: Ламінат, Лінолеум, Керамічна плитка, MDF, Цемент, Сайдинг).

**Кількість** – кількість товару, що знаходиться на складі, проданого чи надходження (діапазон: 1-1000).

**Накладна №** – номер накладної з постачання (приклад: 128).

**Дата** – дата реалізації (приклад: 18.09.2005).

**Ціна** – ціна товару (приклад можливого значення: 50 гривень.).

При обстеженні предметної області виділено атрибутивний складу, наведений в таблиці 2.1.

Таблиця 2.1 – Безліч атрибутів

|  |  |  |
| --- | --- | --- |
| **№** | **Назва атрибутів** | **Ідентифікатор** |
| 1 | Адрес клієнта | Адр\_клт |
| 2 | Накладна № | Ном\_наклад |
| 3 | Ціна при покупці | Ціна\_пок |
| 4 | Вартість при продажі | Варт\_прод |
| 5 | Кількість проданого товару | Кіл\_прод |
| 6 | Назва товару | Назв\_тов |
| 7 | Код товару | Код\_тов |
| 8 | Код клієнта | Код\_клт |
| 9 | Прізвище клієнта | Прізв\_клт |
| 10 | Ім’я | Ім’я\_клт |
| 11 | Телефон клієнта | Тел\_клт |

Отже ми встановили всі основні атрибути, які повинні міститися в базі даних. Для кожного атрибута було визначено значення. Перерахували імена атрибутів і описали їх короткі характеристики.

* 1. **Використання баз даних в будівельних підприємствах**

При аналізі особливостей функціонування будівельних організацій, підтверджується необхідність повсюдних досліджень і перегляду діючих підходів і методів до процесу проектування інформаційних систем (ІС) будівельних підприємств, бізнес-процесів, а також підходів і методів до проектування баз даних, як підстави будь-якої ІС.

Будь-яке велике підприємство має справу з величезними обсягами інформації, яку необхідно обробляти і аналізувати з метою подальшого використання в технологічному циклі виробництва.

Рішення задач подібного роду неможливо здійснити без використання сучасних ІС інтеграції, обробки і управління даними. Їх застосування сприяє зростанню ефективності роботи підприємств, скорочення витрат, оптимізації чисельності персоналу. Розробляючи такі системи на великих підприємствах необхідно проводити інтеграцію даних з високонавантажених ІС і забезпечувати допустимий час відповіді на запит від користувача [62].

Через значну ціни на створення такого роду систем і специфіки діяльності кожної організації, розробка високонавантажених ІС виконується виключно під замовлення, спираючись на вже наявні готові рішення великих виробників. Ці системи використовуються для здійснення певних завдань і вимог, які встановлені технічними завданнями і практично завжди є унікальні продукти, можливі виключно на конкретному підприємстві, будучи абсолютно марними і неефективними на ругом [57, с. 116].

Фірми, які займаються розробкою ІС, не зацікавлені в створенні універсального підходу до проектування тому, що це істотно зменшить комерційну вигоду від їх діяльності.

Сучасна теорія і практика показують необхідність постійного вдосконалення структури баз даних для будівельних організацій, що посилює ефективність управління діяльністю підприємств. Тому особливої важливості набуває питання рефакторінга баз даних як методу радикальних і серйозних перетворень усталеної структури бази даних будівельних підприємств.

Починаючи з 1960-x років, для роботи з даними вдаються до використання спеціальних програмних комплексів, званих системами управління базами даних (СКБД). Серед функцій систем управління базами даних є фізичне розміщення даних і їх опис; пошук даних; підтримка баз даних в актуальному стані; захист даних від некоректних оновлень і несанкціонованого доступу; обслуговування одночасних запитів до даних від декількох користувачів (прикладних програм) [61, с. 202].

Моделі даних, які знаходяться в базах, мають певну логічну структуру, мається на увазі, що вони представлені деякою моделлю, яку підтримує СУБД. До найважливіших належать наступні моделі даних: ієрархічна; мережева; реляційна; об’єктно орієнтована.

Ієрархічна модель відноситься до однієї з найстаріших і перших систем управління базами даних. Вона являє собою вид деревовидної структури і зручна в роботі з впорядкованою інформацією, але має суттєві труднощі в роботі, що містить складні логічні інформаційні зв’язки. Основними інформаційними одиницями в ієрархічній моделі даних служать сегмент і поле.

Мережева модель – логічна модель даних, яка є розширенням ієрархічної моделі. Вона являє собою вид довільного інтерфейсу. Ефективна реалізація оперативності і показники витрати пам’яті відносяться до плюсів ієрархічної і мережної моделей даних. До мінусів мережевий моделі даних відноситься висока складність і жорсткість схеми, яка побудована на основі її бази даних [73].

Реляційна модель даних отримала назву від англійського терміна Relation - відношення. Дана модель описує деякий набір ознак і родових понять, якими повинні володіти всі конкретні СУБД і бази даних, які ними управляються, якщо вони ґрунтуються на цій моделі.

Об’єктно-орієнтована модель даних зберігає в базі не тільки дані, але і методи їх обробки у вигляді програмного коду. Через складність створення і застосування подібних СУБД, такий напрямок не знайшло особливого поширення.

База даних являє собою один або (частіше) кілька файлів, фізична організація яких сильно відрізняється від логічної. Таблиці можуть зберігатися як в окремих файлах, так і всі разом. І, навпаки, для зберігання однієї таблиці можуть бути використані кілька файлів. Так само існують спеціальні файли, які допомагають підтримувати перехресні посилання і виконувати швидкий пошук інформації [74].

Тому при роботі з базами даних зазвичай застосовуються поняття більш високого логічного рівня: запис і таблиця, без заглиблення у подробиці їх фізичної структури.

У підсумку, база даних являє собою набір таблиць з перехресними посиланнями. Щоб працювати з базою даних: додавати, обробляти, змінювати і видаляти дані в ній, потрібні спеціальні програми, які називаються СУБД.

Головним завданням проектування архітектури бази даних, є робота з різними типами і джерелами інформації. Таким чином, джерела і споживачі інформації служать підрозділом даного підприємства, головна контора холдингу, апарат міністерства і підприємства інших галузей (можливі постачальники і споживачі, державні регламентують органи та ін.). Масштабне об’єднання бізнесу показує, що джерела і споживачі інформації можуть знаходитися в будь-якій географічній точці, де це буде необхідно [79].

Звідси випливають стратегічні для архітектури баз даних і ІС в цілому рішення. Об’єднання вимог до динаміки і різноманітності типів інформаційних потоків, що обробляються в ІС, дають можливість узагальнювати характеристику технологій, які формують архітектуру бази даних в складі ІС:

1. компонентна технологія проектування і перекомплектації предметно-орієнтованих операційних баз даних, яка допускає роботу користувачів через загальні, в тому числі, для сховища даних, інтерфейси; розширена технологія сховища даних, яка інтегрує історичні відформатовані дані, архівні текстові документи, звукові і відеоархіви, а також картографічні дані, і включає в себе кошти оперативної аналітичної обробки даних, необхідні види «дружніх» інтерфейсів, наприклад: гіпертекстовий;
2. відкритість бази даних для включення в неї і отримання з неї інформації з використанням принципів глобальної інформаційної магістралі.

Архітектура відкритих систем, яка розширена методами і засобами компонентного формування: на верхньому рівні це відкритість компонентного проектування баз даних та вільного обміну з джерелами інформації будь-яких зовнішніх систем, на нижньому рівні – технологічна відкритість баз даних на основі стандартів переносимості, інтероперабельності, масштабованості і ін [77].

Розширена технологія інтегрального сховища змушує на новій основі ставити питання про розробку інтегрованої сукупності інтерфейсів користувача, що створює природні умови для роботи з інформацією і функціями незалежно від того, до якого класу збережених даних розробник змушений віднести сьогодні його (користувача) інформацію.

В ІС нові архітектури СУБД стануть визначальним, але не єдиним компонентом інтегруючого програмного забезпечення (в тому числі – проміжного). Монітори транзакцій і процесів, засоби семантичного моделювання та використання понятійних моделей, СУБД:

Незалежні засоби розробки і виконання програм – інші класи компонентів програмного забезпечення, що забезпечують досягнення цієї мети.

Рекомендується збереження незалежності від – СУБД на основі використання стандартів та інструментів, що охоплюють різні СУБД. Відмова від зв’язку з однієї СУБД, відкритість CASE – сховища, можливість розвитку метамоделей, підтримуваних в репозиторії, і застосовуваних до них проектних процедур – це лише мінімальні вимоги до методів та інструментів [82].

Розумно орієнтуватися на САЅЕ-системи, які орієнтовані на можливість паралельного проектування компонентів незалежними розробниками (в тому числі – без використання даної САЅЕ – системи) з подальшою інтеграцією метаінформації.

Засоби розробки додатків повинні задовольняти вимогам мобільності додатків і, одночасно, роботи в середовищі розподіленої бази даних.

Подальший розвиток автоматизованих систем управління (ACY) і автоматизованих систем обробки даних різного призначення, стало початком розвитку нової інформаційної технології, створеної за допомогою застосування методів і засобів штучного інтелекту, таких, як експертні системи і системи підтримки прийняття рішень.

Така інформаційна технологія, за допомогою інтелектуального інтерфейсу, буде покращувати ефективність використання баз даних.

Бази даних будівельних підприємств дозволяють здійснювати завдання по автоматизації управління записами, пов’язані з завданнями планування виробництва, поставок необхідних ресурсів і збуту готової продукції. Такий підхід дозволяє коригувати планові завдання в процесі виробництва в залежності від мінливих умов: відсутність поставки ресурсів, відмови обладнання, змінe замовлення.

Модель інтегрованого інформаційного середовища будівельного підприємства включає в себе наступні три головні компоненти:

1) систему автоматичного проектування, за допомогою якої тендерно-договірний відділ обчислює рентабельність від майбутніх закупівельних процедур;

2) автоматизовану систему управління підприємством, за допомогою якої служби підприємства здійснюють управління виробництвом, продажами, закупками;

3) автоматизовану виробничу виконавчу систему, за допомогою якої керівники та служби управляють виробничим процесом.

Системна інформаційна підтримка цих компонентів здійснюється в єдиному інформаційному просторі, що представляє собою сукупність трьох розподілених баз даних:

1) загальні бази даних про укладених раніше договорах, що проходять і майбутні закупівлі, тендери, контракти, розрахунку їх собівартості з урахуванням актуальної вартості матеріалів і фонду оплати праці і інші операційні витрати, передбачуваного прибутку і вартості непередбачених витрат;

2) загальні бази даних про підприємство, що містять нормативну, довідкову, планову і облікову інформацію, яка буде використовуватися різними службами в їх виробничої діяльності;

3) загальні бази даних про кваліфікацію робітників і необхідності своєчасного підвищення кваліфікації, нормах і нормативах, технологічних процесах, технологічних маршрутах, обладнанні, інструменті, оснащенні, транспорті, кадрах, а також планові та облікові дані, показники діяльності об’єкта, виробничої дільниці, бригади.

Роз’єднана інформаційне середовище являє собою сукупність баз даних, що містять відомості, що створюються і використовуються усіма структурними підрозділами підприємства в процесі їх виробничої діяльності.

У складі загальної бази даних про підприємство виділені наступні сім розділів:

1. маркетинг;
2. економіка та фінанси;
3. постачання;
4. кадри;
5. система якості;
6. бухгалтерія і облік.

Інформація в цих розділах формується і використовується в процесі маркетингових досліджень, економічної, фінансової і виробничої діяльності, а також в управлінні персоналом і якістю виконаних робіт, своєчасного постачання матеріалів та обладнання для виконання робіт, а також ведення оперативного обліку і звіряння з контрагентами.

* 1. **Програмний пакет для створення баз даних Microsoft Access**

Microsoft Access – це реляційна система управління базами даних, яка булу розроблено корпорацією Microsoft. Володіє Величезне переліком функцій, серед яких: механізм пов’язаних запитів, установка зв’язку з зовнішніми базами даних і окремими таблицями. В Access можна створювати код додатків, що працюють з базами даних, завдяки запровадження мови VBA [91].

Бази даних Microsoft Access володіють достатньо кількістю плюсів, розробники дуже часто віддають перевагу використанню в житті і работі саме цей додаток. Плюсами баз Даних Microsoft Access є ті, що в одному файлі може знаходитись велика кількість таблиць і індексів, саме того з такими базами Даних працювати найзручніше – створювати резервні копії, переносити дані на інші пристрої і т.д. Серед плюсів також, що імена полів в такій базі даних можна задавати українськими буквами [66].

У складі забезпечення обробки даних Access присутні модулі і макроси. На відміну від макросів, модулі є більш тонким і потужним засобом створення програмних розширень і прикладних програм в середовищі Access. Ці модулі за своїми функціональними засобами відносяться до таких професійних інструментів розробки, як Delphi або Power Builder. Разом з тим, при використанні модулів, від користувача потрібні навички і кваліфікація програміста, а також уміння користуватися основними принципами об’єктно-орієнтованого програмування. Для програмування в середовищі Access використовується процедурна мову Visual Basic для додатків – VBA(Visual Basic for Appications), в який додані об’єктні розширення і набір елементів SQL [91].

Процес створення в Access програмних розширень передбачає активне і безпосереднє використання підходів об’єктно орієнтованого програмування. Об’єктно-орієнтоване програмування засноване на ідеї, суть якої полягає в тому, що процес побудови програми спирається на застосування фундаментальних сутностей – об’єктів.

Кожен з об’єктів має унікальний набір операцій і властивостей, які він може виконувати. Реалізація взаємодій між об’єктами і контроль за ними покладається на виконуючу середу розробки, в якій фіксується код програми. Тому робота програміста, який застосовує технологію об’єктно-орієнтованого програмування, зводиться до створення об’єктів з описом їх властивостей, реакцій об’єкта на ті чи інші зовнішні дії.

В Access більшість програмних об’єктів збігаються з фізичними об’єктами бази даних: таблиці, форми, звіти. Складові об’єкти, що включають в себе набір більш простих об’єктів, крім «видимих» об’єктів має місце і велика кількість «прихованих» об’єктів, управління якими є тільки з програмних додатків. Access має два типи модулів: стандартні і модулі класу. У стандартні модулі входять функції процедури, що викликаються з будь-якого вікна бази даних.

У таких модулях зазвичай міститься програмний код, який має універсальний характер, і який може застосовуватися в різних місцях цього додатка, а також, в різних додатках і модулях. Модулі класу включають в себе, модулі звітів і модулі форм. Вони мають зв’язок з конкретним звітом або формою.

Слід зауважити, що в ранніх версіях Access вони представляли єдиний інструмент об’єктно-орієнтованого програмування. В цих модулях містяться процедури обробки подій, які запускаються у відповідь на їх виклик в звіті або формі. Процедури обробки подій використовують тоді, коли необхідно управління поведінкою звіту або форми і їх відгуком на деякі події, такі, як натискання кнопки [67].

Найбільш важлива область застосування в Access технології об’єктно орієнтованого програмування полягає в програмуванні доступу до даних інформаційного масиву. Для вирішення цього завдання фірма Microsoft розробила спеціальний інтерфейс – DAO (Data Access Objects). Data Access Objects – це об’єкт, який надає інтерфейс до будь-якого типу баз даних або механізму зберігання [53, с. 65].

Цей шаблон програмування застосовується до багатьох мов програмування, більшості програмного забезпечення, що потребує зберігання інформації.

* 1. **Система управління базами даних Access**

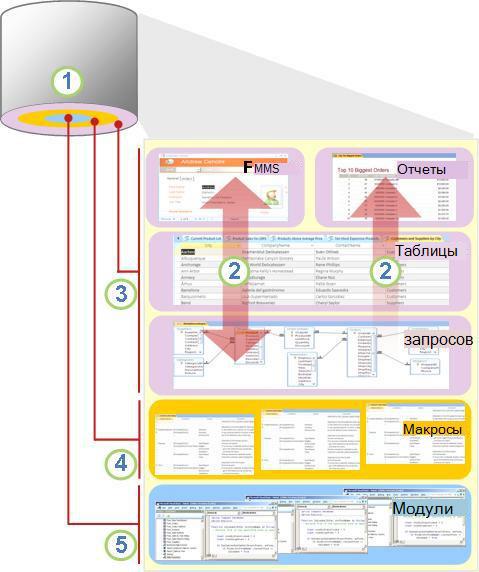
Access – це багата система для роботи з базами даних будь-якого типу на основі реляційної моделі з можливістю динамічного обміну даними з іншими додатками або інтернет-публікаціями. Вона передбачає застосування інструментів автоматизації обробки інформації будь-якого типу, яка представляється в структурованому вигляді. Access є багатофункціональним продуктом. Структура Access включає в себе кілька взаємопов’язаних об’єктів і відрізняється від простого макета електронної таблиці Excel (див. рис. 2.1).

Рисунок 2.1 – Структура Microsoft Access

1. Три способи використання Access: випадковий користувач, досвідчений користувач або розробник.

2. Таблиці, запити, форми і звіти будуються на основі один одного і складають ядро програми бази даних.

3. Випадкові користувачі можуть працювати з майстрами, будівник властивостей, призначеним для користувача інтерфейсом Office Fluent і функціями, подібними Excel, щоб швидко виконувати завдання.

4. Досвідчені користувачі можуть працювати з макросами, областю властивостей, виразами і засобами розробки баз даних, які дозволяють виконувати більш глибокий аналіз і багато іншого.

5. Розробники можуть працювати з модулями і розробляти код VBA, щоб створювати призначені для користувача рішення для баз даних і розгортати динамічні додатки.

Переваги СУБД Access перед другими системами:

* 1. Простий і зрозумілий інтерфейс. До роботи з даною СУБД можна приступати, не володіючи ніякими особливими навичками. Крім того, є можливість скористатися численними довідковими посібниками, а також онлайн-та офлайн-курсами.
  2. СУБД MS Access загальновідома і володіє високою популярністю в настільних ПК. Це означає, що можна знайти безліч інформації по роботі з даною системою.
  3. Копіювання листа Excel в таблицю Access. Можна створити таблицю Access і відобразити її в режимі таблиці, який на даний момент більше схожий на лист Excel.

Врежимі таблиці можна виконати типові завдання по створенню таблиці, такі як визначення типу даних, імені поля або нового поля. Так, якщо ввести дату в порожнє поле, Access задасть для нього тип даних «Дата / час». Якщо вводиться текст, наприклад ім’я, то до поля застосовується тип даних «Текстовий». Щоб перемістити поле, потрібно просто виділити його і перетягнути. Коли дані копіюються з Excel і вставляються в Access, навіть не потрібно попередньо створювати таблицю або відкривати її в режимі таблиці. Access автоматично запитує, чи є в даних заголовки, імовірно визначає типи даних і створює таблицю.

Загальний доступ до даних: зв’язок з Access з листом Excel. Один з найпростіших способів використання можливостей Excel і Access – зв’язування листа Excel з таблицею Access. Access підтримує два принципово різних способу створення таблиць бази даних. Користувачі можуть створювати нові таблиці для зберігання даних в базі даних Access або ж посилання на існуючі дані поза базою даних Access. Дані в пов’язаних таблицях відображаються і поводяться багато в чому так само, як і у власних таблицях програми. Майстер «Диспетчер пов’язаних таблиць» допомагає відслідковувати, знаходити і оновлювати лист Excel або інше джерело даних, якщо він переміщається і зв’язок розривається. При встановленні зв’язку з листом або іменованих діапазоном Excel в додатку Access створюється таблиця, пов’язана з даними Excel. Коли потрібно додати, змінити або видалити дані, зміни вносяться в Excel, а потім виконується оновлення (або повторний запит) даних в таблиці Access. Однак внести зміни в вміст таблиці в Access можна. Дані, пов’язані з Excel, тепер можна використовувати для створення звітів, запитів і доступних тільки для читання форм в Access [55, с. 88]**.**

Переміщення даних: імпорт даних Excel в Access. Якщо потрібно обрізати зв’язок між даними, можна імпортувати дані Excel в Access. У додатку Excel значення слова «імпорт» відрізняється від значення в Access. В Excel «імпорт» (або «підключення») означає встановлення постійного підключення до даних, які можуть оновлюватися. В Access «імпорт» означає одноразовий перенесення даних без постійного підключення до даних. При імпорті даних в додаток Access вони зберігаються у новому або існуючому таблиці без зміни вихідних даних в Excel.

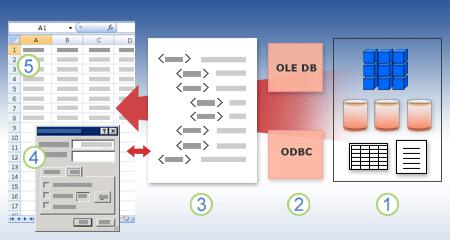
Підключення з Excel до даних Access. Встановивши підключення повторно, можна перенести дані Access назад в Excel. Для цього в Excel необхідно створити підключення (яке часто зберігається в ODC-файлі Office) до бази даних Access і отримати всі дані з таблиці або запиту (див. рис. 2.2). Після підключення до даних, можна автоматично оновлювати книги Excel з вихідної бази даних Access при кожному її оновлення [25].

Рис. 2.2 – Перенесення даних Access в Excel

1. Встановлюється підключення до даних з багатьох джерел.
2. Можна використовувати драйвер ODBC або постачальника OLE DB.
3. Файл підключення містить всі відомості, необхідні для доступу до даних і їх завантаження з Access.
4. Відомості про підключення копіюються з файлу підключення до книги, і їх можна змінити в діалоговому вікні.
5. Дані копіюються в книгу, після чого їх можна використовувати так само, як і дані, які зберігаються безпосередньо в ній.

Використання звітів Access з даними Excel. Після завантаження даних в Access можна скористатися набором засобів для створення і налаштування звітів. Для створення звіту використовується майстер звітів. Для проектування і зміни макета звіту в режимі реального часу з використанням динамічних даних, переміщення блоків даних і зміни їх розміру, додавання і видалення полів з можливістю відразу бачити зміни в міру їх внесення, використовується новий режим макета. Для пошуку, фільтрації і сортування динамічних даних в звіті використовується нове уявлення звіту. Для додавання кнопок команд, номери сторінок, зображення, гіперпосилання і професійних стилів на розсуд використовується безліч майстрів і колекцій елементів управління в групах Елементи управління і Макет розташовані на вкладці Конструктор на стрічці. За допомогою Access є можливість створювати прості звіти, звіти з угрупованням і зведені звіти, поштові наклейки, графічні і підлеглі звіти. Створений звіт можна за допомогою Access поширити в електронному вигляді. Наприклад, можна відправити звіт в електронному листі або зберегти його в певному форматі (такому як знімок Access або PDF-файл), щоб після цього додати на веб-сторінку або сайт SharePoint [27, с. 33].

Використання форм Access з даними Excel. Після завантаження даних в Access можна скористатися набором засобів для створення і налаштування форм. Для створення форми за допомогою декількох клацань мишею використовується майстер форм. Для проектування і зміни макета форми в режимі реального часу з використанням динамічних даних, переміщення блоків даних і зміни їх розміру, додавання і видалення полів з можливістю відразу бачити зміни в міру їх внесення, використовується новий режим макета. Для додавання кнопок команд, списків, полів зі списками, груп параметрів, зображень, масок введення, діаграм, гіперпосилань і професійних стилів на розсуд використовується безліч майстрів і колекцій елементів управління в групах Елементи управління і Макет розташовані на вкладці Конструктор на стрічці. За допомогою Access можна без праці створювати прості форми, форми з вкладками, стрічкові і спливаючі форми, модальні діалогові вікна і підлеглі форми. Процеси створення, зміни і застосування звітів і форм багато в чому схожі, що значно спрощує освоєння програми Access [30].

Фільтрація, сортування і робота із запитами в Access так само, як в Excel. У будь-якому продукті фільтрація використовується для роботи з підмножиною даних, а сортування – для розташування даних в потрібному порядку. В Access можна фільтрувати і сортувати дані в табличному режимі, і для цього не потрібно заново вивчати новий призначений для користувача інтерфейс. Значки, меню команд, команди, умови і діалогові вікна дуже схожі, будь то робота з текстом, числами, датами або порожніми значеннями.

Можна навіть зберігати фільтри і умови сортування разом з режимом таблиці. Можна створити запит в Access, навіть не знаючи, яке SQL-вираз для цього використовується. Чотири майстри запитів допомагають створювати прості запити, знаходити повторення, шукати записи, які не мають відповідностей, і створювати перехресні запити. Не обов’язково навіть створювати запит – за допомогою простої фільтрації і сортування можна надати даними потрібний вид і зберегти його разом з таблицею.

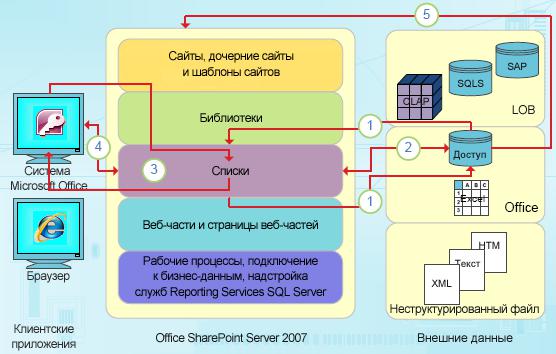
Збір даних за допомогою Access для аналізу в Excel. Завдяки Access процес збору даних від осіб, які перебувають в різних місцях земної кулі, наприклад від співробітників відділу продажів, учасників опитування або представників клієнтів, став набагато простіше. При використанні майстра збору даних Access взаємодіє з Office Outlook, що дозволяє створювати і відправляти повідомлення електронної пошти з формами для введення даних. Обробка відповідей із заповненими одержувачами формами проводиться відповідно до заздалегідь визначених вимог. Наприклад, при виборі автоматичної обробки відповідей вміст форм додається в відповідну таблицю бази даних відразу після їх отримання. Наприклад, можна:

1. провести опитування и зібрати його результати в додатках Access;
2. створити звіти про стан, що містять відомості про обсяги запасів або проблемах, які очікують рішення, які періодічно відправляють співробітники робочої групи;
3. організувати захід, наприклад конференцію або навчальний семінар для збору інформації про контакти, відрядженнях.

Виконання злиття і створення наклейок. Використовуючи таблицю або запит Access в якості джерела даних, можна за допомогою майстра злиття в Microsoft Word створити операцію злиття для листів, повідомлень електронної пошти та конвертів. Якщо потрібно просто створити поштові наклейки, використовується майстер наклейок для створення і друку наклейок з відповідного звіту. Також, можна автоматично додавати відповідний штрихкод для адреси кожного клієнта.

Об’єднання Excel і Access з технологіями SharePoint. В Access, і в Excel є команди для підключення до даних в списках Microsoft Windows SharePoint Services.

Підключення до пов’язаним списками SharePoint в Excel надається тільки для читання (одностороннє), а в Access – для читання і запису (двостороннє). Нові типи даних Access і SharePoint, такі як форматований текст (доданий тільки для підтримки відстежування історії виправлень в поле MEMO), вкладення, лічильник, підстановки і поля, що допускають кілька значень, введені для плавної інтеграції і меншого невідповідності даних (або його відсутності). Після публікації бази даних Access на сайті SharePoint можна навіть завантажити автономну копію даних списку SharePoint в локальну базу даних Access, попрацювати з даними локально, а потім при повторному підключенні до сайту SharePoint передати зміни на сайт. Все конфліктуючі зміни даних, внесені іншими користувачами, будуть оброблятися майстром вирішення конфліктів. Форми і звіти, створені в Access на основі тих же даних, але пов’язані зі списками SharePoint, можна відкривати безпосередньо з Windows SharePoint Services. На рисунку 2.3 представлені можливості інтеграції з SharePoint Server [32, с. 220].

Рисунок 2.3 – Можливості інтеграції з SharePoint Server

* 1. Можна виконувати одноразовий імпорт або експорт даних між уявленнями Access і списком SharePoint.
  2. Можна пов’язувати таблиці Access і список SharePoint шляхом створення постійного двостороннього підключення (оновлені дані списку відображаються в таблиці Access, а оновлені дані таблиці Access - в списку).
  3. Підтримується автономне оновлення даних в Access з подальшою синхронізацією оновлень і розв’язанням конфліктів.
  4. Дані списку можна відображати і змінювати в таблицях, формах і звітах Access.
  5. Підтримується публікація баз даних Access в Office SharePoint Server.
  6. Невисока ціна. Якщо купувати MS Access в складі повного пакету Microsoft Office, то, в порівнянні з іншими платними СУБД, ціна виявиться доступною.
  7. Можливість створення пароля на своїй базі даних.
     1. Вбудована мова VBA високого рівня.

1. Можливість запису макросів. Редактор SQL.

Крім того, Access – це щей пакет, в якому передбачена підтримка елементів ActiveX, що істотно розширює можливості програми в плані того, що вона може використовувати не тільки табличні або текстові компоненти, але і мультимедіа, і об’єкти в інтернеті.

Microsoft Access в більшості випадків використовується для повної автоматизації процесів аналізу якихось процесів в бухгалтерії, бізнесі і т. Д. Завдяки універсальній структурі, програма може усувати появу так званої надмірності даних, коли потрібно змінити якийсь один параметр не шляхом введення нового, а через коригування старого, причому так, щоб його зміна відобразилося у всіх пов’язаних БД [24, с. 98].

СУБД Access надає користувачеві наступні можливості:

1. створювати бази даних;
2. додавати нову інформацію в створені БД;
3. оновлювати або яким-небудь ще чином змінювати дані в БД;
4. видаляти наявну інформацію з БД;
5. дійснювати перегляд даних у вигляді звітів, форм, різних вибірок та

запитів;

1. здійснювати організацію даних за допомогою сортування і / або класифікації інформації;
2. забезпечувати загальний доступ до таблиць і даними при використанні звітів, електронних листів, Інтернету і / або локальної мережі;
3. здійснювати різні види зв’язків між таблицями;
4. створювати кнопкові форми і призначені для користувача інтерфейси управління базою даних всередині СУБД.

Кожна таблиця бази даних має рядки, що включають дані про об’єкти, а також стовпці, завдання яких – визначити їх характеристику. Інша назва рядків - записи, що складаються з полів, які заповнюються даними. Поле обов’язково повинно мати тип даних (числовий, текстової, дата, МЕМО і т. Д.). Всі записи в таблиці містять однаково певні, але по-різному заповнені інформацією поля [11].

Поля в СУБД MS Access задають структуру бази даних, а також задають властивості даними з осередків записів.

Основні властивості такі:

1. Ім’я поля. Зазвичай задають заголовки стовпчикам, тобто називають характеристику об’єкта. Також на ім’я поля до нього можна звертатися за інформацією при виконанні автоматичних операцій з БД.
2. Тип даних. Задає відповідне властивість, визначає, який тип може бути записаний в клітинку.
3. Розмір. Встановлює максимальну довжину записуваних даних. Це властивість не обов’язково використовувати, хоча воно і спрощує роботу, накладаючи обмеження на користувача БД.
4. Формат. Форматує дані поля.
5. Підпис. Виконує ту ж функцію, що і ім’я.
6. Якщо підпис не вказувати, заголовком буде служити перший пункт. Другий можливістю імені – зверненням до нього з БД – підпис не має.
7. Обов’язкове поле. Якщо поставити галочку навпроти цієї властивості, залишити осередок незаповненою буде не можна. Для ключових полів це властивість включено за замовчуванням.

Поля баз даних в Microsoft Access можуть мати такі типи даних, як:

1. Текстовий. Найпростіший тип поля. Незважаючи на назву, може містити як літери, так і цифри, символи та інше. Тому обмеження має тільки по довжині не більше 255 знаків. Зручно для використання, а то й потрібно подальших розрахунків між осередками, тому, якщо в поле вводяться тільки числа, краще використовувати відповідний йому тип.
2. Поле МЕМО. Той же текстовий тип, який зберігає великі обсяги інформації (До 64 кбайт, тобто до 64 тисяч символів). Через цю характеристики МЕМО не може бути використаний в якості ключа або індексу.
3. Числовий. Цифрове поле, що має підтипи, вибір яких залежить від бажаної точності обчислень і т.д. До 8 байт або 16 – для кодів реплікації.
4. Лічильник. Поле, яке не потребує заповнення – значення (числа за зростанням), автоматично заноситься в таблицю, дозволяючи даними в осередках залишатися унікальними.
5. Простіше кажучи, лічильник нумерує записи в базі даних. Зручно використовувати в якості ключа 4 байта, 16 – для кодів реплікації.
6. Логічний. Використовується тільки для значення – 0 (немає) і мінус 1 (так). Можна налаштувати різні варіанти запису – галочка або вибір вручну (розмір – 1 байт).
7. Дата час. Назва типу даних говорить сама за себе. Висновок в поле даних можливий в семи різних форматах. 8 байт.
8. Грошовий. Визначає значення валют. Даний тип даних з’явився для запобігання округлення в обчисленнях. Також 8 байт.
9. Поле об’єкта OLE. Приймає об’єкти різних форматів – графічні, аудіо і т. д. Розмір – до одного гігабайта.
10. Гіперпосилання. Адреси інтернет-сторінок, сайтів. До 64 тис. символів.
11. Майстер підстановок. Заздалегідь передбачає пов’язані таблиці. Дозволяє вибрати значення з іншої таблиці або з поля зі списком. При цьому тип обраного значення встановлюється автоматично. Має розмір первинного ключа. Не більше 4 байт.

Бази даних СУБД MS Access мають у таблицях одне головне – ключове поле. За замовчуванням воно, як уже було сказано, обов’язково до заповнення. Крім того, на нього накладається необхідність бути унікальним, що означає, що вже введене значення в ключовому полі не можна буде ввести в ключовому полі іншому записі цієї ж таблиці. При необхідності можна додати додаткове ключове поле, з менш суворими правилами – унікальність вибирається розробником БД. За допомогою ключових полів здійснюють зв’язку між таблицями бази даних [4, с. 212].

База даних СУБД Access може містити таблиці, які взаємодіють між собою. Для цього використовуються зв’язку, які бувають наступних видів:

1. Один до одного. Це означає, що кожного запису з першої пов’язаної таблиці відповідає одиному запису із другої.
2. Один до багатьох. Кожного запису з першої таблиці відповідає декілька (Дві або більше) з другої.
3. Багато до одного. Безліч записів з першої таблиці пов’язані з однією з другий.
4. Багато до багатьох. Безлічі записів першої таблиці відповідає безліч другий.

Microsoft Access ставить обмеження на імена полів і елементів управління, вони ж діють і на об’єкти СУБД Access:

1. ім’я не повинно містити більше 64 символів;
2. не можна використовувати точку, знак оклику, надрядковий символ або квадратні дужки;
3. ім’я не може починатися з пробілу;
4. не можна використовувати в імені керуючі символи (коди з 0 до 31 в ASCII);
5. ім’я не може включати прямі лапки.

Що ж стосується зручності роботи, воно досягається завдяки наявності основних елементів, які відіграють найважливішу роль при аналізі і обробці даних БД. Серед головних елементів виділяють наступні:

Таблиці є основним об’єктом. Вони задають структуру всієї бази даних.

У них зберігаються дані, які можна змінювати, видаляти або додавати. Таблиці можуть бути взаємопов’язані. На основі цього об’єкта будуються всі інші, за допомогою них же проводяться і основні операції з даними.

Таблиця складається з полів і записів, Кожен запис включає дані про один елемент. Кожне поле містить дані про один аспект елемента таблиці, наприклад ім’я або адресу електронної пошти. Поля часто називають стовпцями або атрибутами.

База даних може включати безліч таблиць, в яких зберігаються дані з різних тем, а кожна таблиця може містити велику кількість полів з даними різного типу, включаючи текст, числа, дати і гіперпосилання.

Можна створювати зв’язку з різними зовнішніми джерелами даних, такими як інші бази даних, текстові файли і книги Excel. Access може використовувати зв’язку з зовнішніми даними, як якщо б вони були таблицями. Залежно від зовнішнього джерела даних і способу створення зв’язку можна змінювати дані в пов’язаної таблиці і створювати відносини з нею. Однак змінювати структуру зовнішніх даних за допомогою зв’язку не можна.

У таблиць і полів є властивості, за допомогою яких можна управляти їх характеристиками і поведінкою. У базі даних Access властивостями таблиці називаються атрибути, що визначають її зовнішній вигляд і роботу. Властивості таблиці задаються на сторінці властивостей таблиці в Конструкторі. Наприклад, можна задати для таблиці властивість Режим за замовчуванням, щоб вказати, як вона повинна відображатися за замовчуванням [53, с. 47].

Властивість поля застосовується до певного поля в таблиці і визначає його характеристики або певний аспект поведінки. Можна задати деякі властивості поля в режимі таблиці. Також, можна налаштовувати будь-які властивості в «Конструкторі» за допомогою області «Властивості поля».

У кожного поля є тип даних, який визначає дані, які можуть в ньому зберігатися (наприклад, великі обсяги тексту або вкладені файли).

Тип даних є властивістю поля, однак він відрізняється від інших властивостей:

* 1. Тип даних поля задається на бланку таблиці, а не в області Властивості поля.
  2. Тип даних визначає, які інші властивості є у цього поля.
  3. Тип даних необхідно вказувати при створенні поля.

Не дивлячись на те, що в кожній таблиці зберігаються дані по певній темі, відомості в різних таблицях бази даних зазвичай пов’язані між собою. Так як дані по різним темам зберігаються в окремих таблицях, їх необхідно якось пов’язати, щоб можна було легко комбінувати дані з різних таблиць. Для цього використовуються зв’язки. Мобільний зв’язок – це логічне відношення між двома таблицями, засноване на їх загальних полях. Поля, що формують зв’язок між таблицями, називаються ключами. Ключ зазвичай складається з одного поля, однак може включати і кілька. Є два види ключів: первинний ключ і зовнішній ключ. Відповідність значень між полями ключів є основою зв’язку між таблицями [24, с. 113].

Роздільне зберігання даних в зв’язаних таблицях забезпечує наступні переваги:

1. узгодженість. Кожен елемент даних заноситься тільки один раз в одну таблицю, тому ймовірність появи неоднозначних або неузгоджених даних знижується.
2. ефективність. Зберігання даних в одному місці дозволяє заощадити місце на диску. Крім того, дані з невеликих таблиць витягуються швидше, ніж з великих. А також, якщо не зберігати дані з різних тем в різних таблицях, виникають порожні значення, що вказують на відсутність даних, або надлишкові дані, що може привести до неефективного використання місця і зниження продуктивності.
3. простота. Структуру бази даних легше зрозуміти, якщо дані з різних тем знаходяться в різних таблицях.

Запити. Ці об’єкти можуть бути вибіркою по будь-якою ознакою з однієї або декількох таблиць, також за допомогою них можна сортувати або фільтрувати інформацію, аналізувати дані, отримувати та надавати користувачеві в зручному вигляді. Серед інших цілей використання запитів можна відзначити:

1. швидкий пошук певних даних шляхом фільтрації із застосуванням певних критеріїв (умов);
2. обчислення або зведення даних;
3. автоматизоване управління даними, наприклад регулярний перегляд актуальних даних.

Особливість запитів полягає в тому, що беруть інформацію з базових таблиць і створюють на їх основі тимчасову результуючу таблицю, яка не має аналога на жорсткому диску, це тільки образ відібраних полів і записів.

У добре структурованої базі даних відомості, які у вас можуть запитати з використанням форми або звіту, найчастіше зберігаються в різних таблицях. Запит може отримати інформацію з різних таблиць і зібрати її для відображення у вигляді форми або звіту, може являти собою звернення до даних для отримання інформації з бази даних або виконання дій з даними. Запити можна використовувати для отримання відповіді на просте запитання, виконання розрахунків, об’єднання даних з різних таблиць, а також для додавання, зміни або видалення даних в таблиці. Це дуже гнучкий інструмент: існує багато типів запитів, і кожен тип створюється з урахуванням завдання [39, с. 127].

Основні типи запитів :

1. Запит на вибірку. Дозволяє переглядати дані тільки з певних полів таблиці або з декількох таблиць одночасно, або ж знаходити дані, які відповідають певним умовам.
2. Повідомлення про порушення. Додавання, зміна або видалення даних. Для кожного завдання існує спеціальний тип запиту на зміну.

Якщо часто потрібно виконувати варіанти певного запиту, можна використовувати запит з параметрами. При виконанні запиту з параметрами у користувача запитуються значення полів, які потім використовуються для створення умов для запиту.

Можна також вказати, якого типу дані дозволяється вводити в якості значення параметра. Тип даних можна налаштувати для будь-якого параметра, але особливо важливо зробити це для числових і грошових даних, а також даних про дату і час.

Коли для параметра зазначений тип даних, користувачі отримують більш зрозумілі повідомлення про помилки в разі введення даних неправильного типу, наприклад введення тексту, коли очікуються грошові дані. Якщо параметр налаштований таким чином, щоб приймати текстові дані, будь введене значення інтерпретується як текст і повідомлення про помилку не відображається [42].

Для створення нової таблиці на основі даних, які зберігаються в інших таблицях, можна використовувати запит на створення таблиці. Для отримання даних з однієї або декількох таблиць і додавання їх в іншу таблицю можна використовувати запит на додавання.

Для зміни даних в таблицях, а також для введення умов, що вказують, які рядки слід оновити, можна використовувати запит на оновлення. Запит на оновлення дозволяє переглянути оновлені дані перед оновленням.

Для видалення даних з таблиць, а також для введення умов, що вказують, які рядки слід видалити, можна використовувати запит на видалення. Запит на видалення дозволяє переглянути видаляються рядки перед виконанням видалення.

Форми. це інструменти для введення і перегляду даних. За допомогою форм можна закрити деякі поля для несанкціонованого введення, можна розмістити спеціальні елементи управління (лічильники, що розкриваються списки, перемикачі, прапорці та ін.) для автоматизації введення. Можна уявити форму за допомогою графічних засобів, у вигляді бланка, якщо введення проводиться з спеціальних бланків. За допомогою форми можна не тільки вводити дані, а й відображати, застосовуючи спеціальні засоби.

Звіти. За допомогою звітів можна переглядати, форматувати і групувати інформацію в базі даних Microsoft Access. Призначені для виведення даних на принтер. У них прийняті спеціальні заходи для групування виведених даних і для виведення спеціальних елементів оформлення, характерних для друкованих документів (верхній і нижній колонтитули, номери сторінок і т.п.).

Звіт – це об’єкт бази даних, зручний для подання її інформацією з однієї з наступних цілей:

1. відображення або поширення зведення даних;
2. архівування моментальних знімків даних;
3. надання відомостей про окремі записах;
4. створення написів.

Макет звіту розбитий на розділи, які можна переглянути в режимі конструктора. Розуміння принципів роботи кожного розділу допоможе створити кращі звіти. Наприклад, від вибору розділу для розміщення обчислюється елемента управління залежить спосіб обчислення результату.

Типи розділів звітів:

1. тема звіту (при друку відображається на початку сторінки). У заголовок повинна містити інформацію, зазвичай поміщається на обкладинці, наприклад емблема компанії, назва звіту або дата. Якщо в заголовку звіту поміщений який вираховується елемент управління, який використовує статистичну функцію Sum, сума розраховується для всього звіту. Заголовок звіту друкується перед верхнім колонтитулом;
2. верхній колонтитул (відображається у верхній частині кожної сторінки). Верхній колонтитул використовується в тих випадках, коли потрібно, щоб назва звіту повторювалося на кожній сторінці;
3. тема групи (відображається на початку кожної нової групи записів). Використовується для друку назви групи. Наприклад, якщо звіт згрупований по виробах, в заголовках груп можна вказати їх назви. Якщо помістити в заголовок групи обчислюваний елемент керування, який використовує статистичну функцію Sum, сума буде розраховуватися для поточної групи. Тема групи може складатися з декількох розділів в залежності від доданих рівнів групування;
4. область даних (відображається один раз для кожного рядка в джерелі записів). В області даних розміщуються елементи управління, складають основний зміст звіту;
5. примітка групи (відображається в кінці кожної групи записів). Примітка групи можна використовувати для друку зведеної інформації по групі. Нижній колонтитул групи може складатися з декількох розділів в залежності від доданих рівнів групування;
6. нижній колонтитул (відображається внизу кожної сторінки). Використовується для нумерації сторінок і для друку посторінковою інформації;
7. примітка звіту (відображається в кінці звіту). Примітка звіту можна використовувати для друку підсумків та іншої зведеної інформації по всьому звіту.

Сторінки. Цей об’єкт дозволяє створити Web-сторінки, компоненти яких здійснюють зв’язок з базою даних. Web-сторінку можна передати клієнту, сама ж база буде розташовуватися на сервері.

Макроси і модулі. Ці об’єкти призначені як для автоматизації повторюваних операцій при роботі з СУБД, так і для створення нових функцій шляхом програмування. Макроси складаються з послідовності внутрішніх команд СУБД Access і є одним із засобів автоматизації роботи з базою. Модулі створюються засобами зовнішнього мови програмування Visual Basic for Applications.

Розробка СУБД в Access виконується за допомогою наступних пунктів:

1. Визначити мета розробки бази даних. Необхідно визначити призначення, дізнатися, в якому напрямку рухатися – як припускають використовувати базу даних, які результати хочуть отримати, які функції повинні бути реалізовані.
2. Визначити приблизну кількість таблиць в базі даних – інформацію необхідно систематизувати і «розкласти все по поличках». Не слід додавати занадто багато полів в одну таблицю: краще розподілити дані на дві і зв’язати їх.
3. Кожна таблиця повинна містити тільки одну тему.
4. Визначити всі поля та їх типи даних в таблицях. Дані в осередках повинні відповідати типам, щоб в подальшому не виникло проблем з обчисленнями, угрупованням і сортуванням.
5. Визначити співвідношення таблиці-поля.
6. Окреслити первинні і (при необхідності) вторинні ключі в таблицях.
7. Побудувати схему даних для БД, що відбиває зв’язки між таблицями.
8. Максимізувати взаємодія між даними за допомогою цих зв’язків.
9. Удосконалити структуру, наочно подивившись на всю наявну інформацію і її можливу систематизацію.
10. Скористатися аналізом самого Access для чергової перевірки.

Створення СУБД у Access можливо двома способами:

1. використання майстра БД для створення необхідних об’єктів;
2. створення порожній БД, доповнюючи її новими об’єктами самостійно.

MS Access дозволяє розширити базу даних вже після її створення, але основну структуру необхідно продумати заздалегідь: такі речі, як типи даних, згодом, особливо після заповнення таблиці, поміняти буде неможливо [67].

Продуману схему даних можна реалізувати за допомогою відповідної вкладки в СУБД. Кожен з видів зв’язку наочно показується в Access. Зв’язки можна видозмінювати або навіть видаляти.

**Висновок до розділу 2**

Отже, у даному розділі випускної роботи було розглянуто характеристику досліджуваного програмного забезпечення Access та його ринкову діяльність, проведено аналіз показників використання баз даних в компаніях. Також був проведений аналіз діяльності, який включає в себе оцінку бізнес-портфелю компаній з використанням ПЗ БД, визначання напрямів вибору та інноваційної діяльності програмного забезпечення Access.

Завершивши дане дослідження, можна зробити висновок, що програмний продукт Microsoft Access займає сильні позиції на ринку програмного забезпечення. Більшість досліджуваної інформації продукту мають досить велику привабливість та позитивні відгуки про конкурентоспроможність, але в той же час дане ПЗ має і деякі слабкі сторони, які відносяться до технічних можливостей продукту.

**РОЗДІЛ 3  
ПЕРСПЕКТИВИ ВИБОРУ ТА РОЗРОБКА БД В СЕРЕДОВИЩІ MS ACCESS**

* 1. **Обгрунтування вибору СУБД**

Для розробки бази даних будемо звертатись в середовище Microsoft Access 2013, яка підтримувана операційною системою Windows 7 і новіше.

У наш час велику популярність має реляційна СУБД Access, яка входить до складу пакету Microsoft Office. Із середовищем програмування Access поєднуються інші додатки Microsoft Office: Word; Excel; PowerPoint. Access – це багато функціональний додаток, в ньому організована надійна система захисту від несанкціонованого доступу до файлів.

Microsoft Access 2013 потужна система, що не дивлячись на це, для недосвідчених користувачів не склад труднощів и будь-яких складнощів розібратися в можливостях і функціоналі СУБД. Зрозумілій інтерфейс і простота налаштувань, зручне и швидке створення таблиць, форм, запитів, взаємозв’язків з іншими додатками пакета, засоби організації роботи з базами даних і захист інформації – це не повний список переваг і сильних сторін цього додатку.

Основні функції СУБД – це опис структури бази даних, обробка даних і управління ними.

Ключовими факторами у виборі MS Access стали:

1. простота і доступність;
2. простота системи для недосвідчених користувачів;
3. величезна кількість макетів, конструкторів, надбудов;
4. зручний інструментарій;
5. можливість редагування в режимі таблиць.
   1. **Аналіз практики та застосування СУБД Access для автоматизації бізнес-процесів**

Для автоматизації обліку договірних відносин був розроблений прототип програмного продукту за допомогою системи управління базами даних Access. Перед тим, як почати створювати базу даних, необхідно побудувати модель предметної області, виділити в ній об’єкти, визначити їх атрибути і встановити якісь зв’язки існують між цими об’єктами [30].

Після того, як були виділені об’єкти, в середовищі Microsoft Access була реалізована схема даних, на якій показуються відносини (зв’язку) між цими об’єктами. Але для того, щоб схема даних була побудована вірно, потрібно визначити ставлення одного об’єкта до іншого заздалегідь:

1. Об’єкти «Замовники» і «Замовлення клієнтів». Ставлення – один до багатьох, так як за один замовник може мати кілька замовлень.
2. Об’єкти «Замовлення клієнтів» і «Товари». Ставлення – один до багатьох, так як за однин товар замовник може оформити безліч замовлень.
3. Об’єкти «Продажі» і «Накладні». Ставлення – один до багатьох, так як за однією продажею замовнику може бути виписана кілька документів про виконаня замовлення.
4. Об’єкти «Товари» і «Склад». Ставлення – один до багатьох, так як один чи декілька товарів може бути вилучена або додана один або більше разів.
5. Об’єкти «Контакти» і «Замовники». Ставлення – один до багатьох, так як по одному контакту можуть міститись різні контактні дані або виконані кілька замовлень підряд.

Ставлення один до багатьох, так як по одній операції замовнику може бути видано кілька документів про виконані роботи субпідряду.

Об’єкти «Замовлення клієнтів» і «Накладні». Ставлення – один до багатьох, так як по одному документу про виконані роботи може бути кілька платіжних доручень.

Після того, як були визначені необхідні об’єкти і відносини між ними, можна приступити до створення безпосередньо самої бази даних. Для цього необхідно, щоб на комп’ютері була встановлена СУБД Microsoft Access. При першому відкритті системи відкриється вікно, де потрібно вибрати: відкрити нову базу даних або вже існуючу (див. рис. 3.1).

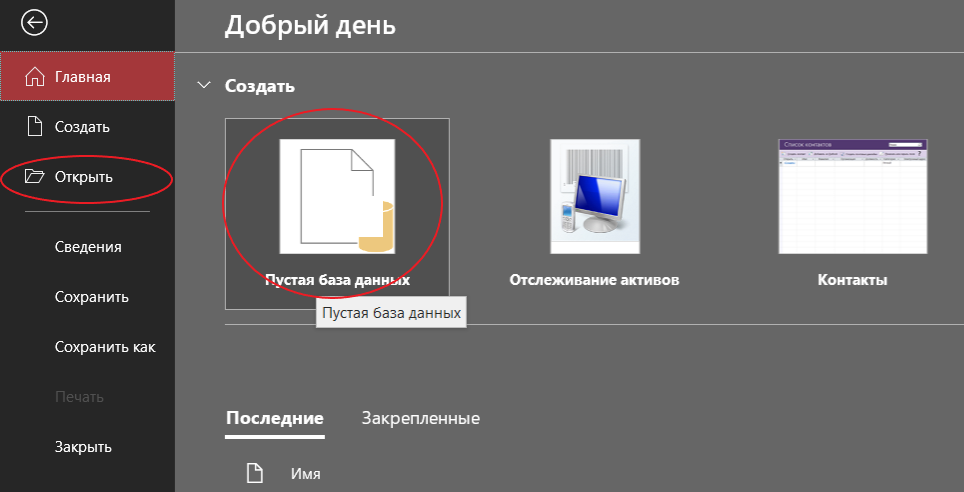


Рисунок 3.1 – Головне вікно Microsoft Access

В першу чергу створюються таблиці, які були перераховані на початку даного пункту. Тому після того, як була відкрита нова база даних, потрібно в верхньому меню програми натиснути на значок «Таблиця» або «Конструктор таблиць» (див. рис. 3.2).

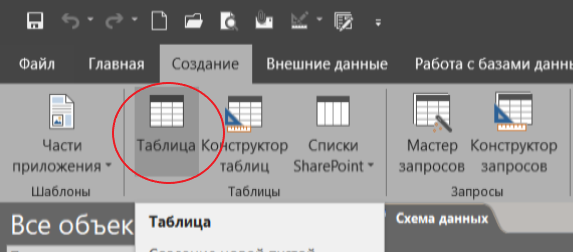


Рисунок. 3.2 – Верхнє меню Microsoft Access

Перший об’єкт, який був створений – таблиця «Контакти» (див. рис. 3.3). Вона складається з трьох стовпців:

1. номер телефону – містить контактні дані замовників. Також це ключове поле. Воно створюється автоматично за замовчуванням. За допомогою цих полів встановлюються зв’язки між таблицями;
2. електрона пошта – містить додаткові контактні дані які замовник надає за бажаням;
3. адрес – містить адресні контактні дані замовників.

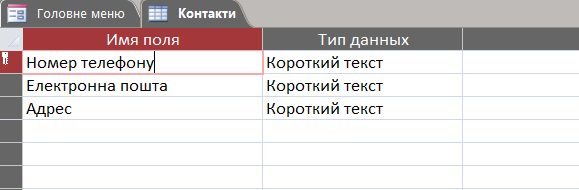


Рисунок 3.3– Таблиця «Контакти»

Також редагуємо введення даних в Номер телефону щоб вони вже містили код країни. В режимі конструктора таблиць в полі тип даних, знизу в таблиці Загальні вибираємо маску вводу і вписуємо тел. +380"000\-00\-00\-00, тепер код країни стоїть автоматично, і при введені наступного номеру ми з економимо час набору«Контакти» (див. рис. 3.4).

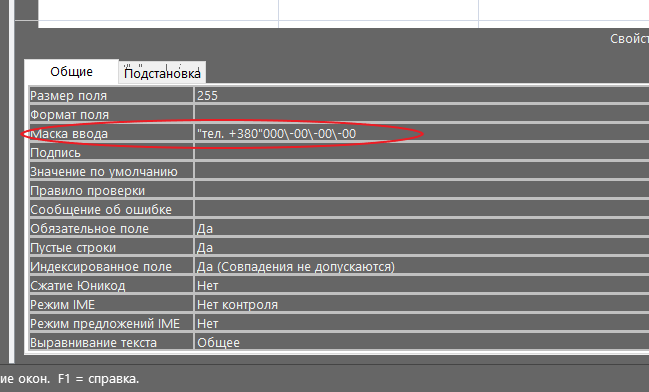


Рисунок 3.4 – Шаблон маски введення контактів до таблиці

Другий об’єкт, який буде необхідний – це таблиця «Замовники» (див. рис. 3.5). У цій таблиці крім ключового поля, номера телефону та дати укладення, присутній ще і поле «код замовника,», за допомогою якого встановлено зв’язок між цією таблицею і таблицею «Замовлення клієнтів».



Рисунок 3.5 – Таблиця «Замовники»

Наступна таблиця – таблиця «Замовлення клієнтів» (див. рис. 3.6). У цій таблиці є 6 полів:

1. ключове поле;
2. поле з кодом товару,;
3. поле з датою;
4. поле з сумою,
5. назва товару;
6. поле з кодом замовлення замовника, яке пов’язує таблицю і цю таблицю «Замовлення».

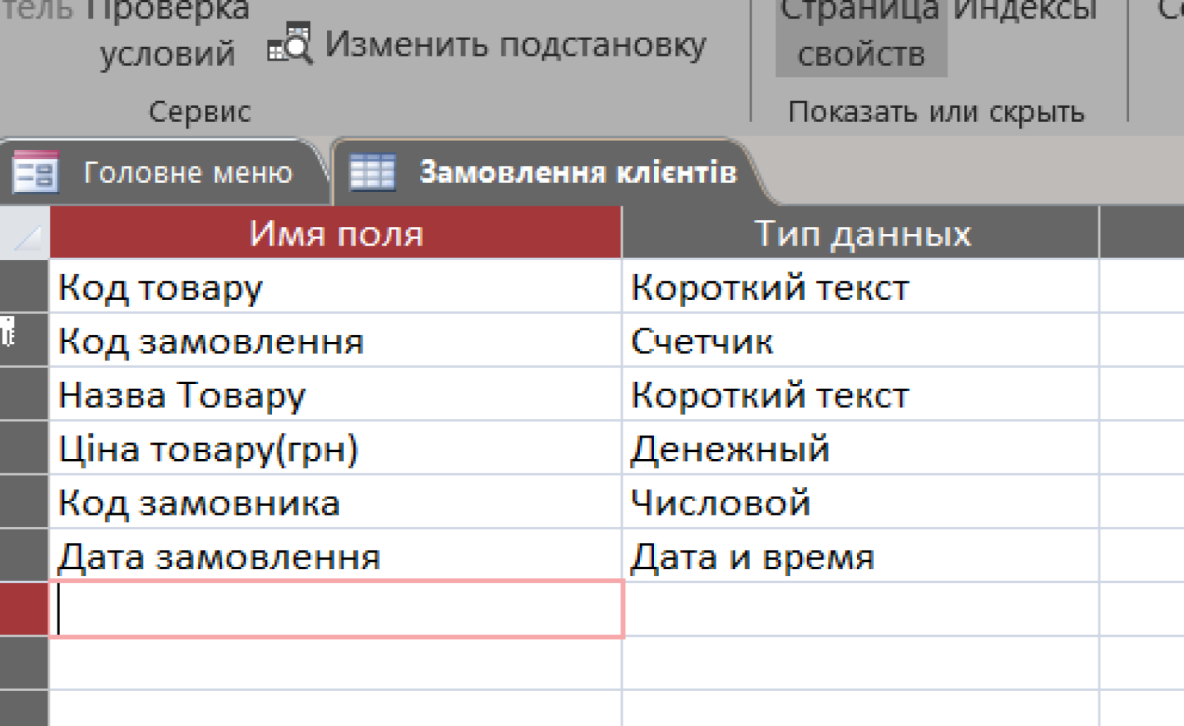


Рисунок 3.6 – Таблиця «Замовлення клієнтів»

Далі був створений об’єкт – таблиця «Товари» (див. рис. 3.7). Ця таблиця пов’язана з таблицею «Накладні» і таблицею «Склад». У ній є такі поля:

1. ключове поле;
2. поле з кодом товару,
3. поле з характеристикою;
4. поле з сумою;
5. назва товару.

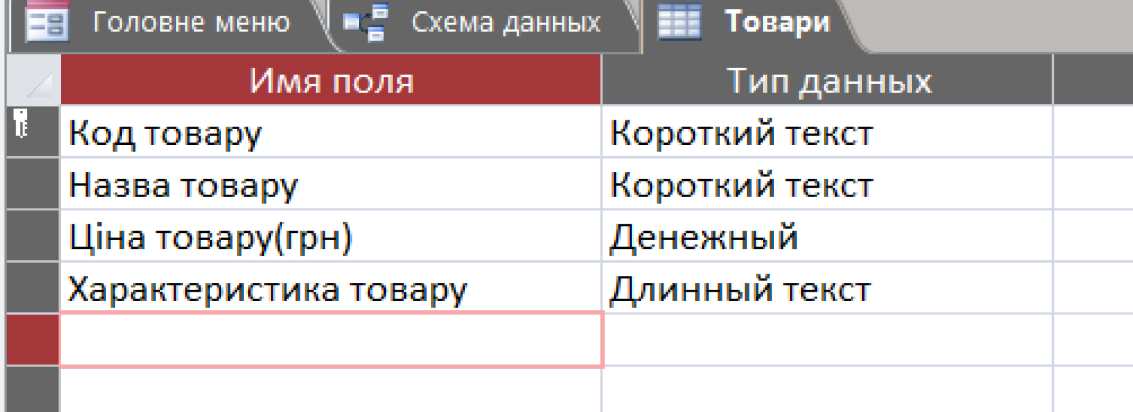


Рисунок 3.7 – Таблиця «Товари»

Наступна таблиця – це «Продажі», містить дані про всі торгівельні операції.

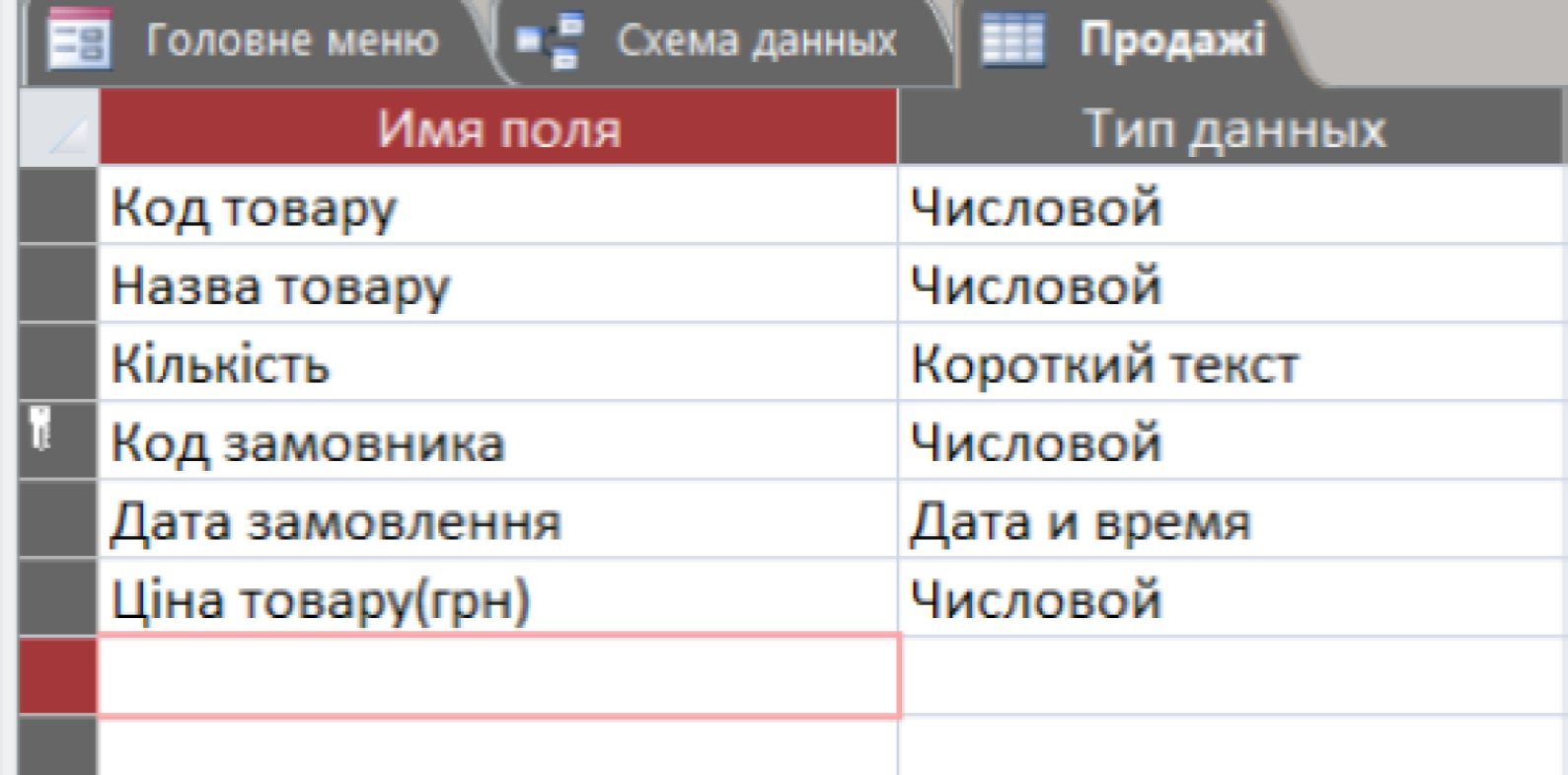


Рисунок 3.8 – Таблиця «Продажі»

У таблиці «Продажі» присутні такі поля, як:

1. ключове поле;
2. код замовника;
3. дата замовлення;
4. сума;
5. код товару;
6. назва товару;
7. кількість.

У таблиці «Накладні» створені поля:

1. ключове поле;
2. код замовлення;
3. поле з сумою;
4. номер телефону;
5. код замовника;
6. ім’я клієнтів;
7. прізвище клієнтів;
8. код товару.

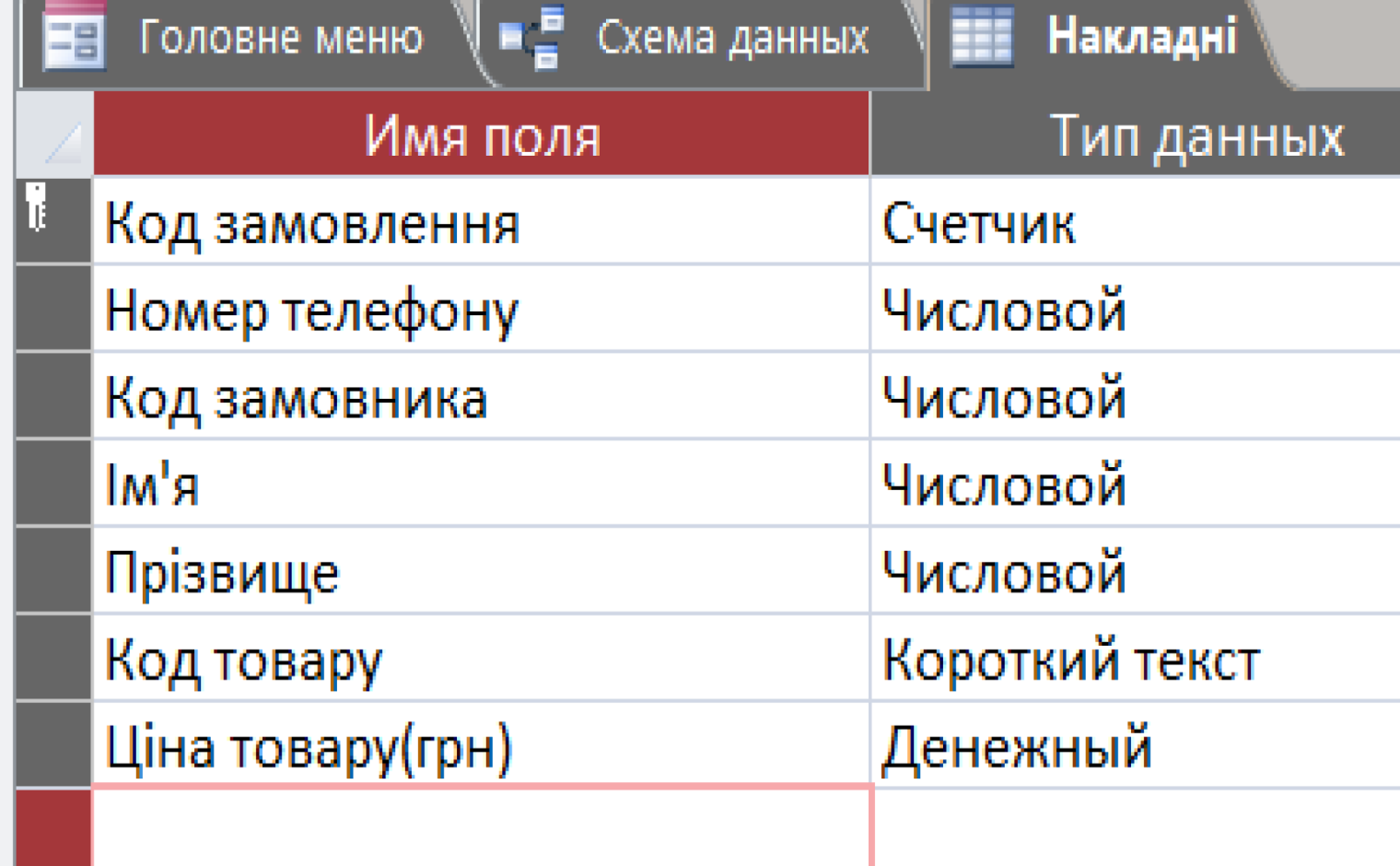


Рисунок 3.9 – Таблица «Накладні»

Накладна – документ, що використовується при роботі з товаром. Не є товаророзпорядчим документом. Цей основний документ що використовується при транспортуванні товарів або абовантажів, який регулює відносини між перевізником, відправником та одержувачем вантажу оформляє та засвідчує договір перевезення вантажу. Приклад налкданої створює саме підприємсто, тому вони можуть відрізнятись, але підходять під одні стандарти [160, с. 92].

І останні необхідні таблиці, які були створені – це таблиця «Склад. Таблиця «Склад» включає в себе такі поля, як: код товару, кількість товару, наявність товару, яке пов’язує цю таблицю з таблицею «Товар».

Після того, як всі таблиці були готові, система автоматично сформувала схему даних, на панелі інструментів Access нажимаємо «Схему даних» і відкриваєтся вікно на якому графічно відображена логічна структура бази даних (див. рис. 3.10, 3.11).

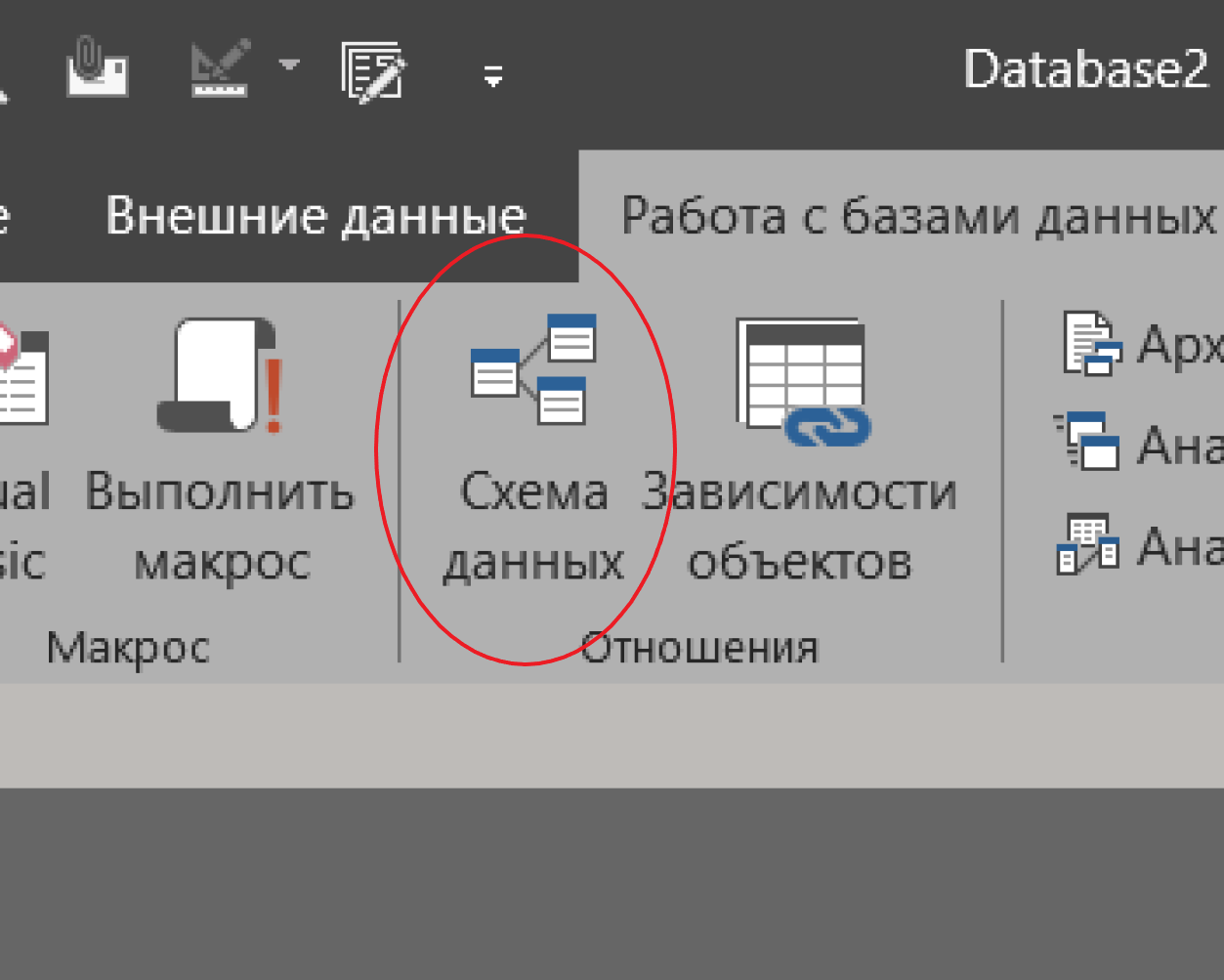


Рисунок 3.10 – Створення схеми даних

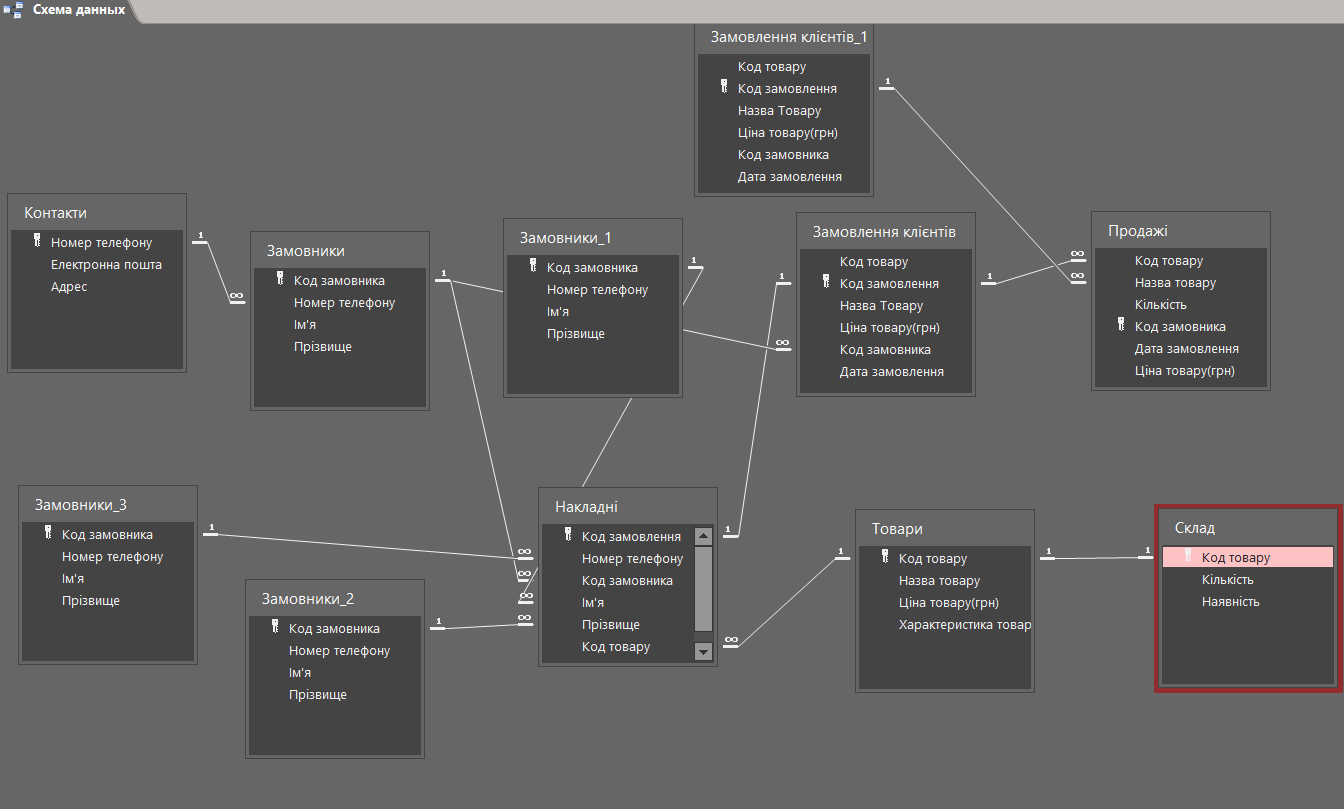


Рисунок 3.11 – Зв’язки у схемі даних

Налаштовуємо зв’язки між таблицями. Після налаштування обов’язково перевіряємо роботу БД.

Наступним кроком було створення форм, призначених для того, щоб користувачам було зручно вводити нові дані в існуючі таблиці і редагувати вже наявні дані.

Першою була створена форма «Замовники». Вона була сформована на основі таблиці «Замовники» (див. рис. 3.12). При створенні даної форми були використані такі інструменти, як:

1. майстер форм;
2. текстове поле;
3. текстове поле зі списком і кнопки.

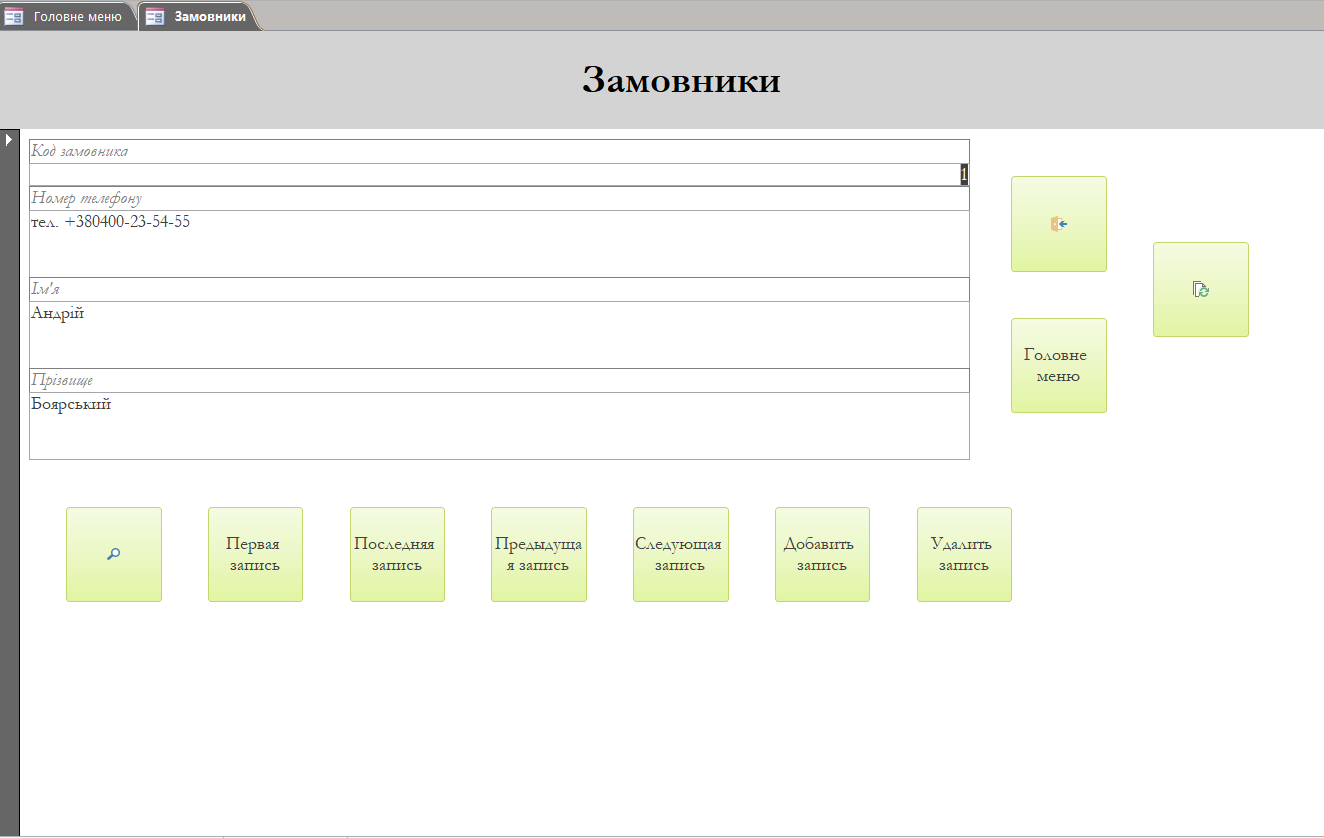


Рисунок 3.12 – Форма «Замовлення»

Наступна форма «Замовлення клієнтів» була створена на основі наявної таблиці «Замовлення клієнтів» (див.рис. 3.13). В даній формі використовувалися наступні інструменти: текстове поле, текстове поле з списком, що випадає, кнопки, поле з вибором дати.



Рисунок 3.13 – Форма «Замовлення клієнтів»

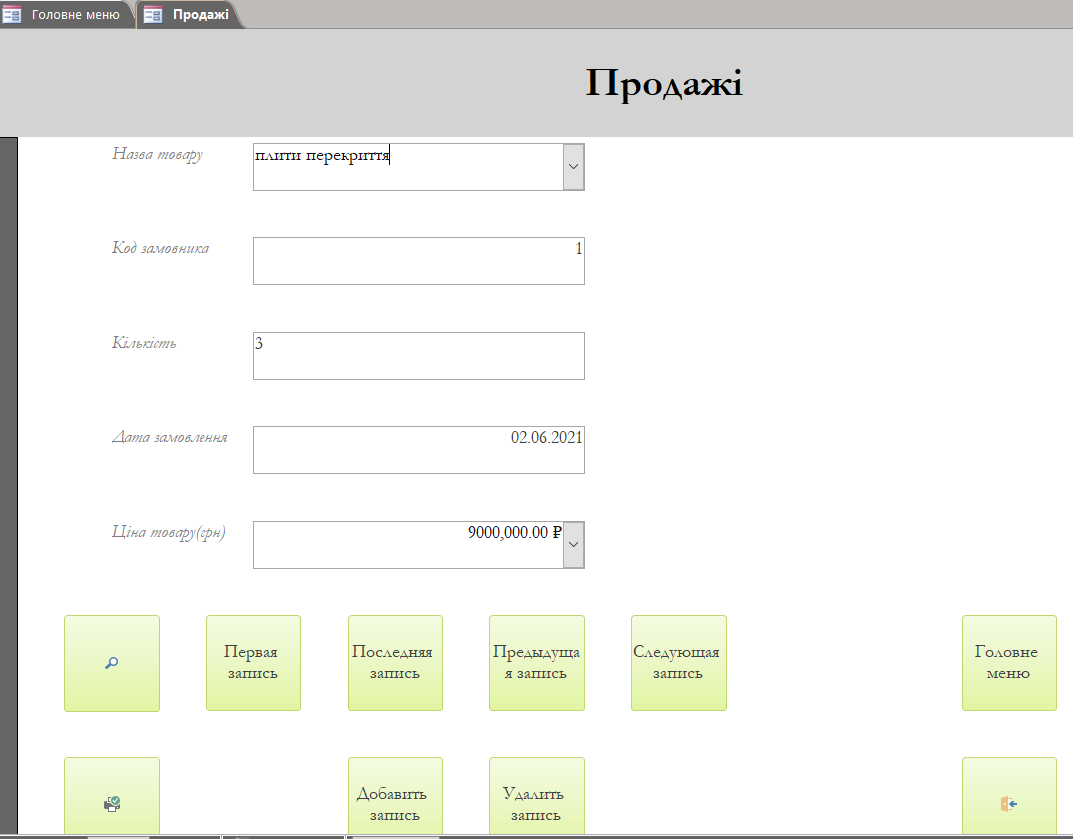


Рисунок 3.14 – Форма «Продажі»

Після цього була створена форма «Продажі», за допомогою якої користувач зможе вносити дані про оплату, яка надходить від замовника (див.рис. 3.15). В даній формі були використані інструменти такі, як: текстове поле і текстове поле з списком, що випадає.

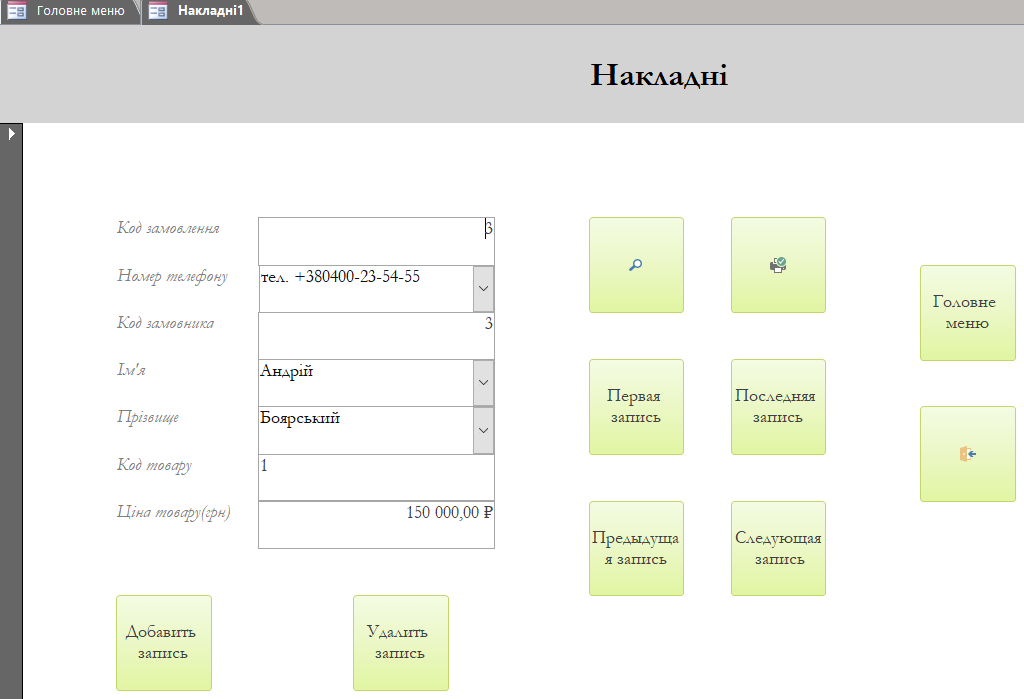


Рисунок 3.15 – Форма «Накладні»

Разом з попередньою формою створена ще одна табличная форма «Накладні», в яку користувачеві буде досить легко і зручно вносити інформацію про оплати, які були зроблені субпідряднику (див. рис. 3.16). У цій формі також були використані: текстове поле і текстове поле з списком, що випадає.

Далі була сформована форма «Контакти» (див. рис. 3.17). За допомогою форми можна вносити додаткові контактні дані клієнтів крім номеру телефонів. Підпорядкована форма «Замовники» яка запозичує дані – номери телефонів. Були використані інструменти: текстове поле і текстове.

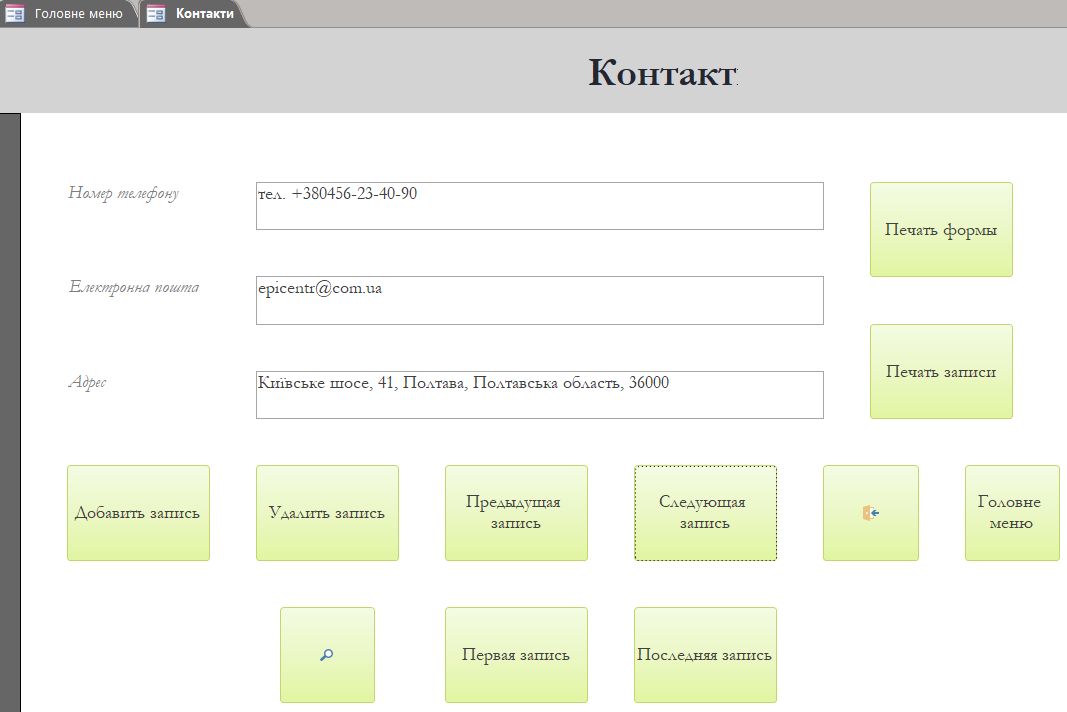


Рисунок 3.16 – Форма «Контакти»

Аналогічно попереднім формі були створені форми «Товар» «Склад» що допоможуть контролювати наявність та характеристики товарів (див.рис. 3.17, 3.19). Використані при створенні інструменти: текстове поле.

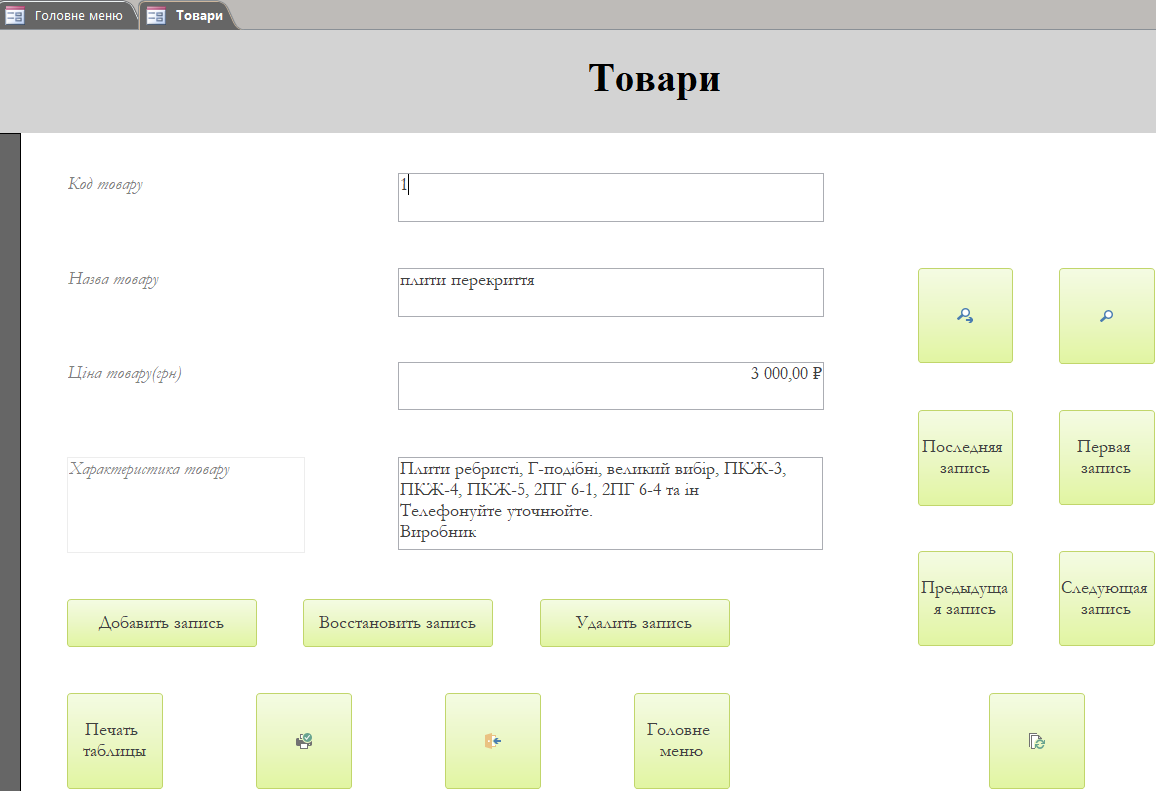


Рисунок 3.17 – Форма «Товари»

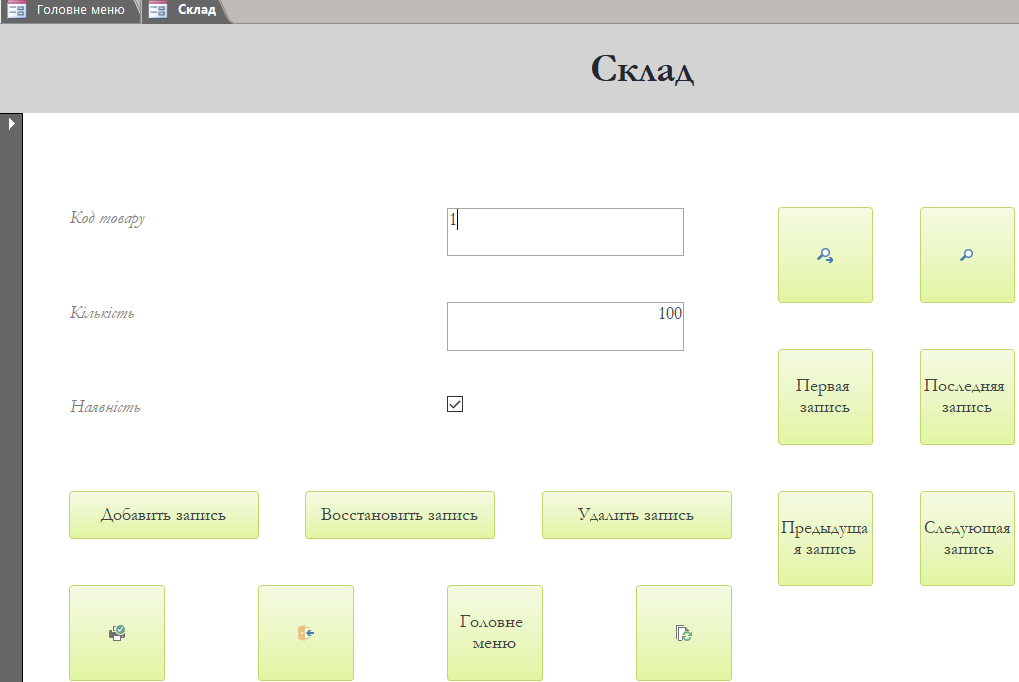


Рисунок 3.18 – Форма «Склад»

Наступний кроком посля створення таблиць і форм є створення форми «Головного меню», що об’єднає всі інші фірми. Меню потрібне для простоти роботи з БД, щоб користувач не відчував дискомфорту при роботі з формами. Форма містить кнопки для миттєвого переходу по інших формах (див. рис. 3.19).

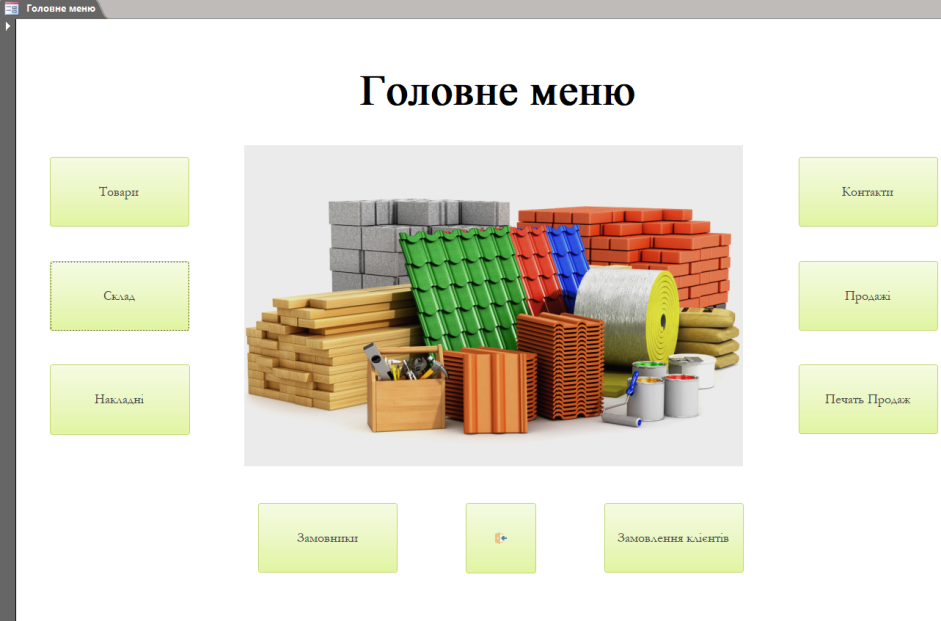


Рисунок 3.19 – Форма «Головне меню»

Також створимо макрос «Autoexec» для автоматичного запуску і завантаженні меню користувача при відкритті файлу БД щоб користувач не витрачав час на ручний запуск меню (див. рис. 3.20). Щоб створити макрос переходимо на панель інструментів Access, вкладка «Створення» і кнопка «Макрос». В відкритому вікні «Макрос» вибираємо «Новий Макрос», «Виділити об’єкт» і заповнюємо поля.



Рисунок 3.20 – Макрос «Autoexec»

Коли всі ці дії були виконані, знову було відкрито головну форму в звичайному режимі (див. рис. 3.21).

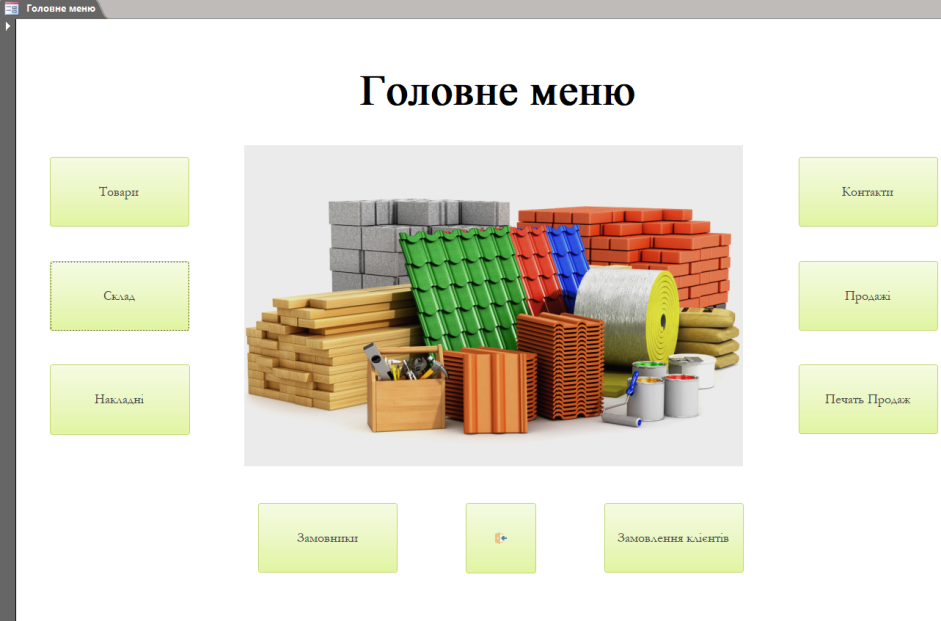


Рисунок 3.21 – Головна форма

Також, знову були протестовані всі кнопки, щоб переконатися, що вони функціонують. Після натискання на кнопки відкриваються сторінки з посиланнями на форми та таблиці. Все працює стабільно.

**Висновок до розділу 3**

Отже, у даному розділі кваліфікаційної роботи бакалавра було виділено, що основним шляхом для удосконалення бізнес діяльності компанії AmoGusBuilding є введення інноваційних технологій.

Нами було впровадженоно нові технології у вигляді БД. Також було представлено інтерфейс користувача в виді форм, та кнопок інтерфейсу що має інноваційне технологічне рішення.

Аналіз сприйняття розробленої технології шляхом оцінки відповідності його характеристик інтересам компанії товаровиробника показав, що даний продукт має успішні шанси впровадження.

**ВИСНОВКИ**

Кваліфікаційна робота бакалавра на тему «РОЗРОБЛЕННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ІТ-ІНФРАСТРУКТУРИ ПОСТАЧАННЯ БУДІВЕЛЬНИХ МАТЕРІАЛІВ» складається з трьох розділів.

У першому розділі роботи ми розглянули теоретичні основи БД та СКБД. Підводячи підсумки до даного розділу можна сказати, що технологічні інновації мають велике значення у діяльності компанії, тому що для утримання ринкових позицій фірма повинна не просто випускати продукт, а й постійно оновлювати свій технічний арсенал. Це є ключем для підвищення якості бізнес процесів та зменшення ресурсозатратності.

В процесі роботи над матеріалом було виявлено, що інноваційна діяльність є важливим фактором у ІТ-Інфраструктурі компанії. Нововведення впливають на конкурентоспроможність та позиції фірми у сфері її функціонування. Кожній компанії слід займатися технологічною інноваційною діяльністю коли її потребує підприємство та споживач. Для визначення доцільності інновацій виробник має проводити аналіз ринку технологій, який вкаже, що більш вигідно.

Важливе місце у діяльності фірми посідає інноваційний процес, який характеризується створення, контролем та впровадженням нових технологій на підприємство. Цей процес є безперервним і діє після впровадженням, адже нововведення має здатність удосконалюватися та набувати нових властивостей.

У другому розділі дипломної роботи було розглянуто характеристику досліджуваного ПЗ та статистику використання його на підприємствах, проведено аналіз технологічних показників ПЗ

Завершивши дане дослідження, можна зробити висновок, що компанія AmoGusBuilding займає сильні позиції на ринку постачання будівельних матеріалів. Більшість порівняних стратегічних бізнес-одиниць не мають тих технологічних рішень як у досліджуваної компанії ,тому теоретично можна прогнозувати досить велику привабливість та конкурентоспроможність, але в той же час компанія має і слабкі ланки, які потребують додаткових інвестицій та введення інновацій, які допоможуть зберегти та покращити стан даного бізнесу.

Третій та завершальний розділ дипломної кваліфікаційної роботи бакалавра був відведений на створення ПЗ БД та впровадження інновації в діяльність компанії AmoGusBuilding. Компанія не повинна втрачати свої позиції на ринку та мати перспективи на майбутній розвиток. Впровадження в даний бізнес допоможе автоматизувати бізне-процеси, зберегти інвестиційне фінансування та зробити його «перезапуск». Було виділено, що основним шляхом для удосконалення бізнес діяльності досліджуваної компанії є введення інноваційного технологічного рішення в ІТ-Інфраструктуру підприємства.

Нами було створено програмне забезпечення у вигляді бази даних та програмного пакету MS Access.

Аналіз сприйняття розробленого ПЗ шляхом оцінки відповідності його характеристик інтересам технологічних контрагентів товаровиробника показав, що даний продукт має достатні шанси на успіх.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Ардерсен Б. Бизнес-процессы. Инструменты совершенствования / Б. Ардерсен. – М.: Стандарты и качество, 2007. – 272 с.
2. Бутчер, Стефан программы лояльности и клубы постоянных клиентов / Стефан Бутчер. – М.:Вильямс, 2004. – 272 с.
3. Гайдамакин Н.А. Автоматизированные информационные системы, базы и банки данных. Вводный курс: учебное пособие./ Н.А. Гайдамакин. – М.: Гелиос АРВ, 2012. – 368 с.
4. Гурвиц, Г. Microsoft Access 2010. Разработка приложений на реальном примере/ Г. Гурвиц – СпБ.: БХВ-Петербург, 2014. – 496 с.
5. Елиферов В.Г. Бизнес-процессы: Регламентация и управление:  
   учебник / В.Г. Елиферов. – М.: НИЦ ИНФРА-М. 2013. – 319 c.
6. Ершов О.Г. О предмете договора строительного подряда / Бюллетень нотариальной практики, 2008, № 4
7. Золотова, С.И. Практикум по Access. / С.И. Золотова. – М.: Финансы и статистика, 2015г. – 109 с.
8. Калянов Г.Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов: учебное пособие / Г.Н. Калянов. – М.: Финансы и статистика, 2009. – 240 с.
9. Кошелев В.Е. Access 2007. Эффективное использование / В.Е. Кошелев. – М.: Бином-Пресс, 2001. – 198 с.
10. Мещеряков, А.Д. Хомоненко. – CП6.: БХВ-Петербург, 2001. – 560 с.
11. Мещеряков, Е.В. Публикация баз данных в Интернете /   
    Е.В. Мещеряков
12. Попов А.И. Свободные инструменты проектирования информационных систем: учебное пособие. / А.И. Попов. – Архангельск: ИПЦ САФУ, 2012. – 74 с.
13. Риз, Дж. Облачные вычисления / Дж. Риз. – Санкт-Петербург: БХВ – Петербург, 2011. – 288 с.
14. Рудакова О.С. Реинжиниринг бизнес-процессов: учебное пособие для студентов вузов / О.С. Рудакова. – М.: ЮНИТИ-ДАНА, 2013. –   
    343 c.
15. Серёдкин, А.Н. Информационные технологии. Создание базы данных в Microsoft Office Access 2007 / А.Н. Серёдкин – ПензГТУ (Пензенский государственный технологический университет), 2013. – 224 с.
16. Сутягин А.В. Договоры в строительстве с комментариями / А.В. Сутягин. – М. РОСБУХ, 2008. – 160 с.
17. Кузин А.В. Базы данных / А.В. Кузин, С.В. Левонисова. – М.: Академия, 2012. – 320 с.
18. Баранова, И.В. Создание баз данных в СУБД ACCESS: учеб.пособие к практ. занятиям / И.В. Баранова, В.В. Быкова; Сибирский федеральный университет. – Красноярск: ИПК СФУ, 2008. – 229 с.
19. Долганова О.И. Моделирование бизнес-процессов: учебник / О.И. Долганова, Е.В. Виноградова, А.М. Лобанова. – М.: Юрайт. 2016. – 281 с.
20. Монахов, Д.Н. Облачные технологии. Теория и практика / Д.Н Монахов, Н.В. Монахов, Г.Б. Прончев, Д.А. Кузьменков. – М.:MAKC Пpecc, 2013. – 128 с.
21. Репин В.В. Бизнес-процессы. Моделирование, внедрение, управление / В.В. Репин. – М.: Манн, Иванов и Фербер, 2013. – 512 c.
22. Репин В.В. Процессный подход к управлению. Моделирование бизнес-процессов. / В.В. Репин, В.Г. Елиферов. М.: Манн, Фербер. 2013. – 544 с.
23. Анодина Н. Н. Документооборот в организации: Практическое пособие / Н. Н. Анодина. М.: Омега-Л, 2009. С. 176
24. Астахова А.В. Информационные системы в экономике и защита информации на предприятиях-участниках ВЭД: Учебное пособие для студентов вузов, обучающихся по направлению подготовки таможенное дело / Под ред. А.В. Астаховой. СПб: Троицкий мост, 2014. С. 216
25. Баласанян В.Э. Электронный документооборот – основа эффективного управления современным предприятием // Секретарское дело. 2002, №2
26. Бобылева М.П. Вопросы использования элементов электронного документооборота внутри организации // Секретарское дело. 2003, №2
27. Бобылева М. П. Эффективный документооборот: от традиционного к электронному / М.П. Бобылева. М.: Издательство МЭИ, 2004. С. 49
28. Гайдамакин Н.А. Автоматизированные информационные системы, банки и базы данных. Вводный курс: Учебное пособие / Под ред. Н.А. Гайдамакина. М.: Гелиос АРВ, 2009. С. 368
29. Глик Д.И. Национальные стандарты в области электронного документооборота // Секретарское дело. 2006, №9
30. Дронов В.А. Система автоматизации делопроизводства и электронного документооборота // Делопроизводство. 2009, №2
31. Есипов В. Е. Экономическая оценка инвестиций: теория и практика / Под ред. В. Е. Есипова. СПб.: Вектор, 2006. С. 287
32. Жеребенкова А.В. Документооборот на предприятии /   
    А.В. Жеребенкова. М.: Вершина, 2005. С. 384
33. Заботина Н.Н. Проектирование информационных систем: Учебное пособие для студентов вузов, обучающихся по специальности прикладная информатика и другим экономическим специальностям / Под ред. Н.Н. Заботиной. М.: Инфра-М, 2013. С. 331
34. Исаев Г.Н. Информационные технологии: Учебное пособие / Под ред. Г.Н. Исаева. 2-е изд. М.: Омега-Л, 2013. С. 464
35. Казакова М. Классификация и примеры современных методов защиты: Учебное пособие / Под ред. М. Казаковой. Ижевск: ИГТУ, 2010. С. 112
36. Касьянова Г.Ю. Документооборот. Основные средства /   
    Г.Ю. Касьянова. М.: АБАК, 2010. С. 256
37. Кузубов В.Н. Принципы построения автоматизированных информационных систем. Техническое и рабочее проектирование АИС: Учебное пособие для вузов / Под ред. В.Н. Кузубова. М.: СГУ, 2009. С. 102
38. Лапшина С.Н. Архитектура предприятия: Учебное электронное текстовое издание / Под ред. С.Н. Лапшиной. Екатеринбург: УрФУ, 2012. С. 110
39. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам: Учебное пособие / Под ред. Ю.А. Маглинец. М.: Бином, 2014. С. 200
40. Максимович Г.Ю. Современные информационные технологии хранения информации и организация доступа // Секретарское дело. 2005. №1
41. Несветаев Ю.А. Экономическая оценка инвестиций: Учебное пособие/ Ю. А. Несветаев. 3-е изд. М.: МГИУ, 2006. С. 161
42. Панасенко С.П. Защита документооборота в современных компьютерных системах // Информационные технологии. 2001. №4
43. Саттон М. Корпоративный документооборот. Принципы, технологии, методология внедрения / М. Саттон. М.: БМикро, 2002. С. 446
44. Степанов Е.А. Информационная безопасность и защита информации: Учебное пособие / Е.А. Степанов. М.: Инфра-М, 2001. С. 304
45. Титаренко Г.А. Автоматизированные информационные технологии: Учебное пособие / Под ред. Г.А. Титаренко. М.: ИНФРА-М, 2009.   
    С. 121
46. Тихонов В.И., Юшин И.Ф. Электронные архивы и электронный документооборот // Отечественные архивы. 2010. №2
47. Чернов В.Н. Системы электронного документооборота / В.Н. Чернов. М.: РАГС, 2009. С. 412
48. Шелухин О.И. Моделирование информационных систем: Учебное пособие для студентов вузов, обучающихся по специальностям сети и системы коммуникации, многоканальные телекоммуникационные системы / Под ред. О.И. Шелухина. 2-е изд. М.: Телеком, 2012. С. 536
49. Шишов О.В. Современные технологии и технические средства информатизации: Учебное пособие для студентов вузов, обучающихся по направлению сервис / Под ред. О.В. Шишова. М.: Инфра-М, 2014. С. 462
50. Бабич В.Н. Инновационная модель бизнес-процесса: Учебное пособие для студентов, обучающихся по направлению подготовки информационные системы и технологии / Под ред. В.Н. Бабича, А.Г. Кремлева. Екатеринбург: УрФУ, 2014. С. 184
51. Белов В.В. Проектирование информационных систем: Учебник для студентов, обучающихся по направлению прикладная информатика и другим экономическим специальностям / Под ред. В.В. Белова, В.И. Чистякова. М.: Академия, 2013. С. 352
52. Глущенко П.В. Модели систем электронного документооборота на предприятии / Под ред. П.В. Романова. М.: Вузовская книга, 2011. С. 120
53. Гухман В.Б. Основы защиты данных в Microsoft Office: Учебное пособие / В.Б. Гухман, Е.И. Тюрина. 1-е изд. Тверь: ТГТУ, 2005. С. 100
54. Емельянова Н.З. Основы построения автоматизированных информационных систем: Учебное пособие / Под ред. Н.З. Емельяновой, Т.А. Партыка. М.: Инфра-М, 2011. С. 416
55. Панов А.В. Разработка управленческих решений при помощи информационных технологий: Учебное пособие для студентов ВПО / Под ред. Т.Н. Ананьевой. 2-е изд. М.: Телеком, 2014. С. 151
56. Бодров О.А. Предметно-ориентированные экономические ИС: Учебник для студентов вузов, обучающихся по направлению и специальности Прикладная информатика / Под ред. О.А. Бодрова, Р.Е. Медведева. М.: Телеком, 2013. С. 244
57. Внуковский Н.И. Экономическая оценка эффективности инвестиций на предприятии: Учебное пособие для студентов, обучающихся по направлениям прикладная информатика, менеджмент, экономика / Под ред. Н.И. Внуковского, Г.П. Бутко. Екатеринбург: УрФУ, 2013. С. 258 Гадасин В.А. От документа – к электронному документу. Системные основы / Под ред. В.А. Гадасина, В.А. Конявского. М.: РФК Имидж Лаб, 2006. С. 189
58. Грекул В.И. Управление внедрением информационных систем: Учебник для студентов вузов направления подготовки бизнес-информатика / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина. М.: Бином, 2014. С. 224
59. Сафонов В.О. Платформа облачных вычислений Microsoft Windows Azure: Учебное пособие / Под ред. В.О. Сафонова. М.: Бином, 2013.   
    С. 234
60. Советов Б.Я. Архитектура информационных систем: Учебник для студентов вузов, обучающихся по направлению подготовки информационные системы и технологии / Под ред. Б.Я. Советова, А.И. Водяхо, В.А. Дубенецкого, В.В. Цехановского. М.: Академия, 2012. С. 288
61. Шматалюк А. Моделирование бизнеса. Методология ARIS / А. Шматалюк, М. Ферапонтов, А. Громов. М.: Весть-Метатехнология, 2001. С. 327
62. 1С:Документооборот. Общее описание, [Электронный ресурс], 1998-2019. URL: <http://v8.1c.ru/doc8/>
63. ARIS Express – бесплатная программа для моделирования бизнес-процессов и оргструктуры [Электронный ресурс] // Программные продукты – Электрон. дан. – 2017. – URL: https://bpmsoft.org/
64. ARIS Express [Электронный ресурс] // Технологии – Электрон. дан. – 2017. URL: http://compress.ru/
65. TANAIS / Типовое внедрение СЭД, [Электронный ресурс], 1997-2019.URL:[http://www.tanais.ru/konsalting\_i\_avtomatizatsiya\_biznesa/sistemy\_](http://www.tanais.ru/konsalting_i_avtomatizatsiya_biznesa/sistemy_%20elektronnogo_dokumentooborota/stoimost_vnedreniya_sed/tipovoe_vnedrenie_sed) [elektronnogo\_dokumentooborota/stoimost\_vnedreniya\_sed/tipovoe\_vnedrenie\_sed](http://www.tanais.ru/konsalting_i_avtomatizatsiya_biznesa/sistemy_%20elektronnogo_dokumentooborota/stoimost_vnedreniya_sed/tipovoe_vnedrenie_sed)
66. База данных СУБД Access [Электронный ресурс] // Информационные технологии – Электрон. дан. – 2017. – URL: https://www.syl.ru/
67. Базы данных MS Access [Электронный ресурс] // Программное обеспечение – Электрон. дан. – 2017. – URL: https://http://fb.ru/
68. Базы данных и системы управления базами данных [Электронный pecypc] Режим доступа: https://elibrary.ru/item.asp?id=30661832
69. Булиныш, З.А. Миграция баз данных mysql в облачную среду google cloud sq1 и возможности её применения в сфере образования / 3.А.Булиныш [Электронный pecypc]. – Режим доступа: https://elibrarY.ru/item.asp?id=17924492
70. Диадок. Электронный документооборот с контрагентами, [Электронный ресурс], 2009-2019. URL: [https://www.diadoc.ru](https://www.diadoc.ru/)
71. Достоинства и недостатки СУБД. [Электронный pecypc] – Режим доступа: <http://pivot-taЫe.ru/dostoinstva-i-nedostatki-subd.html>
72. Журнал о системах электронного документооборота ECM-Journal / Выбор системы документооборота, [Электронный ресурс], 2006-2019. URL: <http://ecm-journal.ru/card.aspx?contentid=1630137>
73. Инструменты управления и моделирования бизнес-процессов [Электронный ресурс] // Организация эффективного управления – Электрон. дан. – 2017. – URL: http://rzbpm.ru/
74. Информационные потоки предприятий и методы их исследования [Электронный ресурс] // Бизнес-статьи – Электрон. дан. – 2017. – URL: http://bbcont.ru/
75. Информационный портал «TAdviser» / Обзор российского рынка систем электронного документооборота, [Электронный ресурс], 2005-2019. URL: [http://www.tadviser.ru/index.php/СЭД?cache=no&ptype=system#ttop](http://www.tadviser.ru/index.php/%C3%90%C2%A1%C3%90%C2%AD%C3%90%E2%80%9D?cache=no&amp%3Bptype=system&amp%3Bttop)
76. Использование баз данных и других источников информации на ИТ-зависимость: какую роль играют данные в современном бизнесе [Электронный pecypc] – Режим доступа: http://business.cnews.ru/artic1es/2016-06-16itzavisimostkakuYuго1iцraYut
77. Ковалева Н. Н., Холодная Е. В. Комментарий к Федеральному закону «Об информации, информационных технологиях и о защите информации» от 27.07.2006 №149-ФЗ [Электронный ресурс], Система ГАРАНТ, 2007. URL: <http://base.garant.ru/5366510/#ixzz3QIIaSw5V>
78. Контрактные отношения в строительстве [Электронный ресурс] // Электронная библиотека – Электрон. дан. – 2017. – URL: http://uchebnik.online.ru/
79. Методология проектирования реляционных баз данных с использованием данных табличного вида. [Электронный pecypc] – Режим доступа: https://elibrarY.ru/item.asp?id=19189250
80. Моделирование бизнес процессов [Электронный ресурс] // Менеджмент качества – Электрон. дан. – 2017. – URL: http://www.kpms.ru/.
81. Моделирование бизнес-процессов [Электронный ресурс] // Сервис имитационного моделирования бизнес-процессов – Электрон. дан. – 2017. – URL: https://www.bpsimulator.com/
82. Особенности методологии проектирования информационных систем для малого и среднего бизнеса. [Электронный pecypc] – Режим доступа: https://elibrary.ru/item.asp?id=9481340
83. Особенности методологии проектирования информационных систем для малого и среднего бизнеса. [Электронный pecypc] – Режим доступа: https://elibrary.ru/item.asp?id=9481340
84. Официальный сайт СКБ Контур, [Электронный ресурс], 1997-2019. URL: [https://kontur.ru](https://kontur.ru/)
85. Портал Центрального Банка Российской Федерации / Статистика. Инфляция на потребительском рынке, [Электронный ресурс], 2000-2019. URL: <http://www.cbr.ru/statistics/?Prtid=macro_sub>
86. Развитие моделирования бизнес-процессов [Электронный ресурс]/ Сайт Минкомсвязи России / Какие существуют виды электронной подписи, является ли электронная подпись аналогом подписи на бумаге, [Электронный ресурс], 2001-2019. URL: <http://minsvyaz.ru/ru/appeals/faq/32>
87. Система электронного документооборота Tessa, [Электронный ресурс], 2006-2019. URL: [https://mytessa.ru](https://mytessa.ru/)
88. Система электронного документооборота Дело, [Электронный ресурс], 2008-2019. URL: <https://www.eos.ru/eos_products/eos_delo/>
89. Системы BPM [Электронный ресурс] // Система менеджмента качества – Электрон. дан. – 2017. – URL: http://www.kpms.ru/
90. Совместное предприятие «Интегра» / Основы использования электронной подписи при таможенном декларировании [Электронный ресурс], 2006-2019. URL: [http://spintegra.ru/ed/ed\_basics/ep\_tam\_decl](http://spintegra.ru/ed/ed_basics/ep_tam_decl/)
91. Справка по Access [Электронный ресурс] // Microsoft Office – Электрон. дан. – 2017. – URL: https://products.office.com/ru/
92. Экономические науки – Электрон. дан. – 2017. – URL: http://www.scienceforum.ru/