

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ДО ЗАХИСТУ ДОПУЩЕНА

Голова циклової комісії,
спеціаліст в/к

_____ С.О. Сабанов

ВИПУСКНА РОБОТА МОЛОДШОГО СПЕЦІАЛІСТА

ТЕМА ВИПУСКНОЇ РОБОТИ:

Розробка програми з обліку попередніх та резервних заказів

Виконав

ст. гр.ІПЗ -118К9

В.Т. Керімова

Керівник

викладач

Д.О. Костерной

Запоріжжя

2022

СТРУКТУРНИЙ ПІДРОЗДІЛ «ФАХОВИЙ КОЛЕДЖ ЕКОНОМІКИ ТА
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ ПРАТ «ПВНЗ «ЗІЕІТ»

Циклова комісія з інформаційних технологій

ЗАТВЕРДЖУЮ

Голова циклової комісії,

спеціаліст в/к

_____ С.О. Сабанов

17 січня 2022 р.

З А В Д А Н Н Я
НА ВИПУСКНУ РОБОТУ МОЛОДШОГО СПЕЦІАЛІСТА
студенту гр. ІПЗ-118К9,
спеціальності 121 - «Інженерія програмного забезпечення»

Керімовій Владі Тофіківні

(прізвище, ім'я, по батькові)

1. Тема: «Розробка програми з обліку попередніх та резервних заказів»

затверджена наказом № 09.2 – 19 від 04 березня 2022 р.

2. Термін здачі студентом закінченої роботи: 18 червня 2022 року

3. Перелік питань, що підлягають розробці: _

1. Провести огляд літератури, що присвячена тематиці досліджень

2. Провести бесіду з керівником

3. Розглянути та проаналізувати аналоги

4. Зробити аналіз апаратного обладнання

5. Виконати всі поставлені задачі

6. Протестувати розробку

7. Оформити звіт за результатами роботи

4. Календарний графік підготовки випускної роботи молодшого спеціаліста

№ етапу	Зміст	Термін виконання	Готовність по графіку (%), підпис керівника	Підпис керівника про повну готовність етапу, дата
1	Формулювання (корегування) теми випускної роботи молодшого спеціаліста та збір практичного матеріалу за темою випускної роботи	17.01.22-26.02.22		
2	I атестація. I розділ випускної роботи молодшого спеціаліста	28.03.22-02.04.22		
3	II атестація. II розділ випускної роботи молодшого спеціаліста	10.05.22-14.05.22		
4	III атестація III розділ випускної роботи молодшого спеціаліста, висновки та рекомендації, додатки, реферат	30.05.22-04.06.22		
5	Перевірка випускної роботи молодшого спеціаліста програмою «Антиплагіат»	30.05.22-18.06.22		
6	Доопрацювання випускної роботи молодшого спеціаліста, підготовка презентації, отримання відгуку керівника і рецензії	30.05.22-18.06.22		
7	Попередній захист випускної роботи молодшого спеціаліста	14.06.22-18.06.22		
8	Подача випускної роботи молодшого спеціаліста на кафедру	За 3 дні до захисту		
9	Захист випускної роботи молодшого спеціаліста	20.06.22-25.06.22		

Керівник

Костерной Д.О.

(підпис)

(прізвище та ініціали)

«____» _____ 2022 р.

Завдання прийняв до виконання

Керімова В.Т.

(підпис студента)

(прізвище та ініціали)

«____» _____ 2022 р.

РЕФЕРАТ

Дипломна робота містить 70 сторінок, 4 таблиці, 26 рисунків,

Мета: формування умінь з проектування та реалізації роботи програмного інтерфейсу з використанням бази даних для обраної предметної області на основі отриманих теоретичних знань на протязі усього навчання.

Об'єктом дослідження є реалізовувати програмне забезпечення з спроектованою базою даних для створення заказів по обраній предметній області.

Предметом дослідження є програма з обліку попередніх та резервних заказів

Здійснено детальний огляд предметної області та сучасних аналогів.

Результатом роботи є побудовані компоненти системи, опис процесу розробки цих компонентів: реалізована база даних для збереження необхідної інформації, побудовано інтерфейс користувача для керування системою. Розробка бази даних здійснювалася засобами системи керування базами даних MySQL, а розробка інтерфейсу – мовою C#.

Програмна система, дозволить слідкувати за замовленнями, додавати нові чи роботи резервування необхідного квитка, контролювати процес перельотів, роботи працівників, а також це дозволить дуже просто заносити інформацію про нові літаки, нових співробітників, в якій можливо буде дуже швидко дізнатись про придбані квитки, та нові авіарейси.

СУБД, БД, C#, VISUALSTUDIO, АСУ, SQL

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ
І ТЕРМІНІВ

Скорочення	Повна назва	Пояснення/переклад
СКБД	Система керування базами даних	
БД	База даних	
ER	entity-relationship diagram	Схема «сутність – зв'язок»
ПЗ	Програмне забезпечення	
РМД	Реляційна модель даних	
GB	gigabyte	Гигабайт
АСК	автоматизована система керування	

ЗМІСТ

РЕФЕРАТ.....	4
ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ.....	5
ВСТУП.....	8
РОЗДІЛ 11 АКТУАЛІЗАЦІЯ ТА ДЕТАЛЬНИЙ АНАЛІЗ ЗАВДАННЯ.....	11
1.1Актуальність тематики.....	11
1.2Аналоги програм.....	11
1.3Аналіз предметної області «Аеропорт».....	12
1.4Аналізування та поняття ієрархічної моделі.....	13
1.5Розробка реляційної модель даних.....	14
1.6Поняття бази даних та СКБД.....	17
1.7Концептуальне проектування.....	20
1.8Вибір моделі даних.....	21
1.7 Логічне проектування.....	22
1.8 Фізичне проектування.....	24
1.9Висновок по першому розділу.....	26
РОЗДІЛ 2..... РОЗРОБКА ОБ’ЄКТІВ ДЛЯ ДОСТУПА ДО РЕЛЯЦІЙНИХ ДАНИХ	27
2.1 Поняття реляційних модель даних.....	27
2.2 Маніпулювання даними.....	28
2.3 Висновок по другому розділу.....	31
РОЗДІЛ 3.....	32
ПРОЕКТУВАННЯ БАЗИ ДАНИХ.....	32
3.1. Апаратні вимоги до програмно забезпечення.....	32
3.2. Створення входу на сервер на базі фіксованих серверних ролей.....	32
3.3. Створення ролей рівня баз даних.....	35

	7
3.4. Створення користувачів баз даних.....	35
3.5. Створення завдань резервного копіювання та операторів.....	36
3.6. Використання криптографічних алгоритмів для захисту.....	37
3.7 Фізична реалізація та тестування проекту.....	39
3.8 Висновок по третьому розділу.....	52
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	55
ДОДАТОК А СТВОРЕННЯ ТАБЛИЦЬ SQL.....	57
ДОДАТОК Б ЗАПИТИ SQL.....	66

ВСТУП

Актуальність. Керування літаками в сучасних умовах є дуже складною задачею не лише через обробку великих об'ємів даних, але й через оперативність отримання довідкової або розрахункової інформації, що впливає, насамперед, на успіх компанії чи часу витраченому на перельоти.

Останні події у Світі показали, що обслуговування аеропортів , літаків . чи персоналу аеропорту або пілотів, займає багато часу , я гадаю що цю проблему можливо вирішити завдяки створенню ПЗ до аеропорту , з великою кількістю функціоналу для швидкої роботи співробітників «Аеропорту» . Тому, необхідною умовою ефективного функціонування структури керування будь-якого рівня є наявність діючої інформаційної системи, на яку покладено функції автоматизованого збору, обробки та маніпулювання даними. Така система включає в себе технічні засоби обробки даних, програмне забезпечення та обслуговуючий персонал [5, 9 - 12].

Саме основні із вищенаведених функцій можуть виконуватись системами керування базами даних (БД), які є невід'ємною частиною автоматизованої системи керування аеропортами.

Метою є дослідження та розробка бази даних за ПЗ автоматизованої системи керування (АСК) аеропортом та в майбутньому можливості її застосування в сучасних аеропортах.

Основна частина

Однією із основних функцій АСК є забезпечення накопичення, збереження, пошуку та видачі інформації за запитами користувачів (посадових осіб органів керування) для наступного її використання, обробки, обміну, тощо. Для забезпечення зазначених функцій АСК створюються БД.

Саме в базах даних інформаційних ресурсів АСК циркулює інформація, без якої неможлива колективна робота посадових осіб органів керування, а також вирішення окремих розрахункових задач [4, 9-11]. Тому, очевидно, що базу даних слід вважати основним елементом будь-якої АСК.

Об'єктна модель, що представляє статичну структуру системи, такої АСК повинні входити такі підсистеми [1]:

1. описуючи об'єкти предметної області;
2. база даних, як структуроване сховище інформації про об'єкти;
3. об'єктно-реляційний перетворювач, необхідний для забезпечення автоматичної синхронізації між даними додатку у вигляді наборів пов'язаних об'єктів і даними, що зберігаються в системі керування БД в реляційному вигляді (наприклад, для реляційної моделі даних такими специфікаціями є ERD (Entity Relationship, діаграми “ В склад сутність- зв'язок”);
4. система автоматизованого резервного копіювання БД;
5. підсистема авторизації і керування правами доступу до різної інформації в БД;
6. підсистема керування інформаційними профілями структурних підрозділів для забезпечення всебічної інформації тощо.

Важливою складовою АСК є інформаційне забезпечення, зокрема, база даних, яка в функціональному відношенні пов'язана із іншими підсистемами АСК. Для розробки БД доцільно використовувати сучасні напрацювання теорії баз даних, які використовуються в цивільному секторі, але із відповідною модернізацією, яка враховує специфіку військової сфери.

Основними вимогами до даних, які є в базі, для їх якісної і своєчасної обробки, є [4, 8]: предметна орієнтованість (всі дані в системі керування збираються, очищаються, узгоджуються, доповнюються, пакетуються і подаються в єдиній, зручній для їх аналізу і використання формі); інтегрованість (всі дані в системі керування узгоджені між собою і зберігаються в єдиній базі)

незмінність (вихідні дані, після того, як вони узгоджені, затверджені і внесені в базу даних, залишаються незмінними і використовуються виключно в режимі читання); підтримка хронології (дані хронологічно структуровані і відображають історію за достатній, для виконання задач аналізу і прогнозування, період часу).

Методика побудови інформаційної моделі бази даних АСК передбачає виділення трьох рівнів: інформаційно-логічний, датологічний і фізичні рівні, яким відповідають інфологічна, датологічна і фізична моделі даних. Перша, серед зазначених моделей формується, як правило, науково-дослідними установами, які виконують наукове супроводження розробок, а друга і третя - виробником програмного забезпечення [3].

РОЗДІЛ 1

АКТУАЛІЗАЦІЯ ТА ДЕТАЛЬНИЙ АНАЛІЗ ЗАВДАННЯ

1.1 Актуальність тематики

На сьогоднішній день, усе найактуальнішим стає швидкодія роботи підприємства та бізнесу, і сьогодні в цей не легкий час, потрібно якомога швидше роботи замовлення, треба прискорити час як обробки оформлення замовлення, так і створення обліку попередніх та резервування нових замовлень. Що надасть змогу бізнесу, чи підприємству, з використанням мого програмного забезпечення виконувати швидко свою працю, завдяки чому програма буде конкурентно здатною, виконувати велику кількість замовлення за короткий час.

За метою мого проекту, програма буде створена для обліку попередніх та резервних заказів, із предметної області «Аеропорт», у наступних розділах, я більш ширше розкрию предметну область аеропорту, та наведу інформацію про розробку бази даних, та програмного інтерфейсу до неї

1.2 Аналоги програм

Серед аналогів програмного забезпечення взято українського розробника, з його хмарним сервісом, РемОнлайн, котрий має спільну мету створення, а саме скоротити час на замовлення, вскоротити час роботи співробітників .

РемОнлайн - хмарний сервіс, для роботи з яким потрібен лише інтернет. Просте в освоєнні веб-додаток дозволяє комфортно та ефективно працювати всім співробітникам вже з першого дня використання, а керівникам – контролювати їхню роботу. Нижче на рисунку, зображен сам хмарний сервіс.

The screenshot displays the 'Замовлення' (Orders) section of the RemOnline service. At the top, there are summary cards: '2 замовлення Мої замовлення' (2 orders My orders), '3 замовлення Термінових' (3 orders Urgent), '2 замовлення Прострочені' (2 orders Overdue), and '966,75 грн Чекають оплату' (966.75 UAH Waiting for payment). Below these is a table of orders with columns for ID, status, date, status, creator, client, amount, type, and name.

Замовлен...	Строк замовле...	Статус	Створено	Клієнт	Сплачено	Тип замовлення	Наймену...
A321	-4 д. 06 лют. 2022 00:00	Передано в ЦЕХ, неоплачено	Ольга Савченко 04 лют. 2022 16:48	Скопик Володимир 0000000000	213,75 грн	Основний	Хімчистка, а...
A320	— 06 лют. 2022 00:00	Готове	Ольга Савченко 04 лют. 2022 16:31	Кравченко Олена 0000000000	0 грн	Основний	Хімчистка, а...
A319	-5 д. 05 лют. 2022 00:00	Очікує погодження	Ольга Савченко 04 лют. 2022 10:39	Володимир Тихонюк 0000000000	0 грн	Основний	Хімчистка, а...
A318	— 06 лют. 2022 00:00	Закрито	Ольга Савченко 03 лют. 2022 16:25	Олександр Петроц... 0000000000	792 грн	Основний	Хімчистка, а...
A317	-5 д. 05 лют. 2022 00:00	Неоплачено	Ольга Савченко 03 лют. 2022 12:26	Наталія Кальченко 0000000000	621 грн	Основний	Хімчистка, а...
A316	-7 д. 03 лют. 2022 19:00	В роботі	Ольга Савченко 02 лют. 2022 17:56	Олександр Скрипч... 0000000000	250 грн	Основний	Хімчистка, а...
A315	— 04 лют. 2022 00:00	Готове, неоплачено	Ольга Савченко 02 лют. 2022 10:39	Мар'янна Іванова 0000000000	440 грн	Основний	Хімчистка, а...
A314	-8 д. 02 лют. 2022 00:00	Передано в ЦЕХ, сплачено	Ольга Савченко 01 лют. 2022 17:48	Петенко Богдан 0000000000	200 грн	Основний	Хімчистка, а...
A313	-7 д. 03 лют. 2022 00:00	В роботі, неоплачено	Ольга Савченко 01 лют. 2022 10:29	Юрій Ланда 0000000000	135 грн	Основний	Хімчистка, а...
A312	— 03 лют. 2022 00:00	Закрито	Ольга Савченко 30 січ. 2022 13:30	Куценко Станіслав 0000000000	400 грн	Основний	Хімчистка, а...

Рис. 1.1 – Хмарний сервіс РемОнлайн

Серед не доліків я позначу те що розробка над потрібними мені функціями ще ведеться, та хмарний сервіс, не дає гарантію конфіденційного збереження інформації, а також робить обов'язковим наявність інтернету для роботи, також немає можливості приспособити хмарний сервіс, для себе. На мою думку у програмі для великого підприємства, повинна обов'язкова можливість оновлення функціоналу, та прямого доступу до коду пз.

1.3 Аналіз предметної області «Аеропорт»

Предметна область – частина реального світу, що підлягає вивченню з метою організації керування і, в кінцевому рахунку, автоматизації. Предметна область представляється безліччю фрагментів.

Кожен фрагмент предметної області характеризується безліччю об'єктів і процесів, що використовують об'єкти, а також безліччю користувачів, які характеризуються різними поглядами на предметну область. Побудова інформаційно-логічної моделі даних предметної області, оцінка об'ємів даних БД. На першому етапі

проектування БД АСК повинна бути розроблена концептуальна (інфологічна) модель предметної області, яка базується на інформаційних потребах майбутніх користувачів БД та за своєю сутністю є вихідними даними для виробника інформаційного забезпечення.

Мета інфологічного проектування - створити структуровану інфологічну модель предметної області (ПО), для якої розроблятиметься БД (із врахуванням сучасних технологій роботи з базами даних).

В рамках інфологічної моделі описуються інформаційні об'єкти, їх атрибути та зв'язки між ними, а сама модель розробляється без прив'язки до конкретної системи керування базами даних (СКБД) [6].

Наповнення БД необхідно виконувати поетапно. Кінцева оцінка повинна враховувати об'єми даних, необхідні для зберігання декількох поколінь резервних копій БД, вільні об'єми пам'яті, необхідні для перезапису даних, а також архітектурні особливості БД.

В даній дипломній роботі поставлена задача створення розробка програми з обліку попередніх та резервних заказів, предметною областю якої буде Аеропрот. Тому при її виконанні було проведено дослідження даної предметної області. В рамках даної роботи реалізується проектування і розробка навчальної бази даних предметної області. Отже, отримана БД не буде відображати всю бізнес-логіку даної предметної області, а обмежитися лише її частиною.

1.4 Аналізування та поняття ієрархічної моделі

Ієрархічна модель бази даних складається з об'єктів з покажчиками від батьківських об'єктів до нащадків, сполучаючи разом зв'язану інформацію. Ієрархічні бази даних можуть бути представлені як дерево, що складається з об'єктів різних рівнів. Верхній рівень займає один об'єкт, другий, — об'єкти другого рівня і так далі. Між об'єктами існують зв'язки, кожен об'єкт може включати декілька об'єктів нижчого

рівня. Такі об'єкти знаходяться відносно предка (об'єкт більш близький до кореню) до нащадка (об'єкт нижчого рівня), при цьому можлива ситуація, коли об'єкт-предок не має нащадків або має їх декілька, тоді як у об'єкту-нащадка обов'язково лише один предок. Об'єкти, що мають загального предка, називаються близнятами. Перші системи керування базами даних використовували ієрархічну модель даних, і в часі їх поява переганяє появи мережевої моделі.

Вузол — це сукупність атрибутів даних, що описують деякий об'єкт. На схемі ієрархічного дерева вузли представляються вершинами графа. У мережевій структурі кожен елемент може бути пов'язаний з будь-яким іншим елементом. Мережеві бази даних подібні ієрархічним, за винятком того, що в них є покажчики в обох напрямках, які сполучають родинну інформацію.

Не дивлячись на те, що ця модель вирішує деякі проблеми, пов'язані з ієрархічною моделлю, виконання простих запитів залишається досить складним процесом. Також, оскільки логіка процедури вибірки даних залежить від фізичної організації цих даних, то ця модель не є повністю незалежною від додатка. Іншими словами, якщо необхідно змінити структуру даних, то потрібно змінити і додаток.

Мережева база даних відрізняється більшою гнучкістю, оскільки в ній існує можливість встановлювати додатково до вертикальних ієрархічних зв'язків горизонтальні зв'язки.

1.5 Розробка реляційної моделі даних

Реляційна модель певної предметної області є набором відношень, що змінюються в часі. Поняття реляційний (англ. relation — відношення) пов'язано з розробками відомого англійського фахівця в області систем баз даних Едгара Кодда (Edgar Codd). Ці моделі характеризуються простотою структури даних, зручною для користувача табличною виставою і можливістю використання формального апарату алгебри стосунків і реляційного числення для обробки даних.

РМД – Реляційна модель даних. Відношення між таблицями встановлює зв'язок між даними, які знаходяться у різних таблицях. Зв'язки між таблицями встановлюються шляхом об'єднання співпадаючих значень ключових полів.

Реляційна модель орієнтована на організацію даних у вигляді двовимірних таблиць.

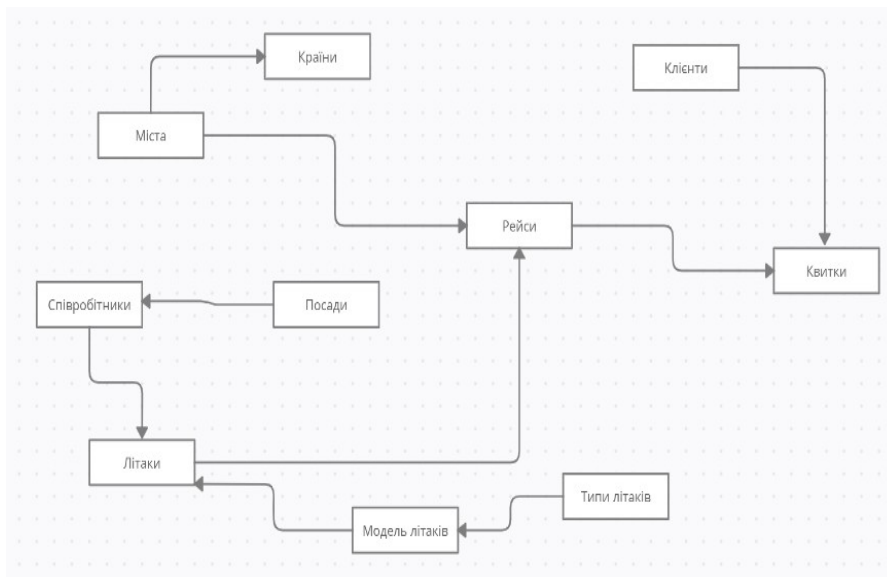


Рисунок. 1.5.1 – Реляційна модель даних

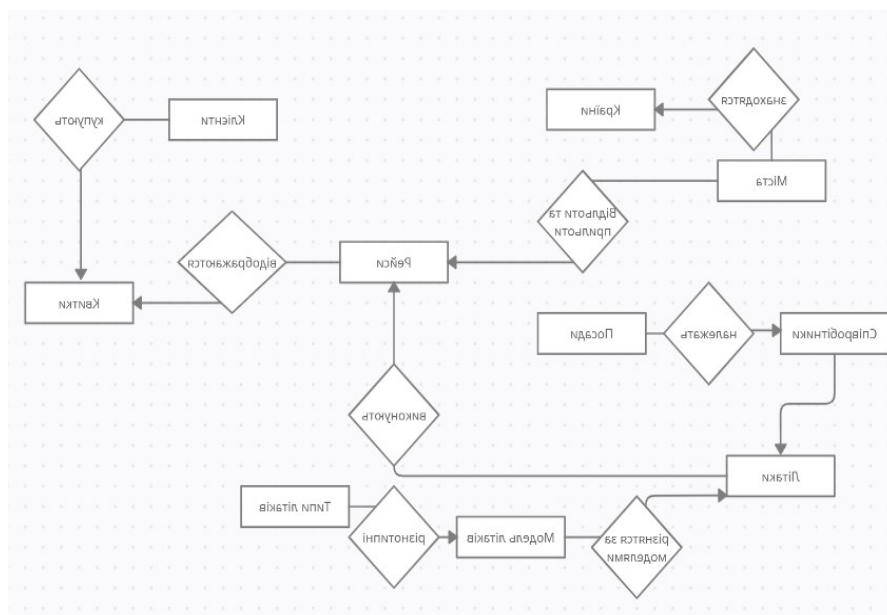


Рисунок. 1.5.2 – Реляційна модель даних з описом

Базовими поняттями (елементами) реляційних СКБД є:

Таблиця-1.1 -Базові поняття СКБД

Атрибут	Заголовок стовпця
Відношення	Таблиця
Домен	Стовпець
Кортеж	Рядок
Первинний ключ	Один або кілька атрибутів
Сутність	Опис властивостей об'єкту
Схема відношення	Рядок заголовків таблиць
Тип даних	Тип значень елементів таблиць

Атрибут – поіменована характеристика сутності.

Відношення – є двовимірною таблицею, що містить деякі дані.

Сутність – містить деякі дані – це об'єкт будь-якої природи. Дані про сутність зберігаються в відношенні таблиці.

Первинний ключ – мінімальний набір атрибутів, за значенням яких можна знайти кожний екземпляр сутності.

Об'єктивно-орієнтовна модель - система керування обробляє дані як абстрактні об'єкти, наділені властивостями, у вигляді неструктурованих даних, і використовуючи методи взаємодії з іншими об'єктами навколишнього світу.

Проектувальники не наповнюють базу конкретними даними (замовник може вважати їх конфіденційним і не надавати стороннім особам). Виняток становить експериментальне наповнення модельними даними на етапі відладки об'єктів бази.

Друга категорія виконавців, що працюють з базами даних, - користувачі. Вони отримують вихідну базу даних від проектувальників і займаються її наповненням і обслуговуванням. У загальному випадку користувачі не мають засобів доступу до

керування структурою бази – лише до даних, да і то не до всіх, а до тих, робота з якими передбачена на конкретному робочому місці.

Відповідно СКБД має два режими роботи: проектувальний і призначений для користувача.

Перший режим призначений для створення або зміни структури бази і створення її об'єктів. У другому режимі відбувається використання раніше підготовлених об'єктів для наповнення бази або здобуття даних з неї. У основі реляційної моделі використовується поняття “відносини”, яке використовується для уявлення набору екземплярів об'єкту (сутність) та відносин (зв'язків) між об'єктами.

1.6 Поняття бази даних та СКБД

База даних — це набір відомостей, що відносяться до певної теми або завдання, такий як відстеження замовлень клієнтів або зберігання колекції звукозаписів. Мета будь-якої інформаційної системи — обробка даних про об'єкти реального світу. Якщо база даних зберігається не на комп'ютері або на комп'ютері зберігаються тільки її частини, доводиться відслідковувати інформацію з цілого ряду інших джерел, які користувач повинен скоординувати і організувати самостійно.

Створюючи базу даних, користувач прагне упорядкувати інформацію за різними ознаками і швидко витягати вибірку з довільним поєднанням ознак. Зробити це можливо, тільки якщо дані структуровані.

Структурування — це запровадження угод про способи подання даних.

Неструктурованими називають дані, записані, наприклад, у текстовому файлі.

Користувачами бази даних можуть бути різні прикладні програми, програмні комплекси, а також фахівці предметної області, що виступають у ролі споживачів або джерел даних, називаються кінцевими користувачами.

У сучасній технології баз даних передбачається, що створення бази даних, її підтримка та забезпечення доступу користувачів до неї здійснюються централізовано

за допомогою спеціального програмного інструментарію — системи керування базами даних. [2]

В якості СКБД для реалізації бази даних була використана Microsoft SQL Server які є настільними СКБД реляційного типу, які мають всі необхідні засоби для виконання перелічених вище функцій. Система керування базами даних Microsoft SQL є одним з найбільш популярних програм в сімействі настільних СКБД. Всі версії SQL Server мають у своєму арсеналі засоби, що значно спрощують введення і обробку даних, пошук даних та надання інформації у вигляді таблиць, графіків і звітів. Починаючи з версії SQL Server 2005, з'явилися також Web-сторінки доступу до даних, які користувач може переглядати за допомогою програми Internet Explorer.

Крім цього, SQL Server дозволяє використовувати електронні таблиці та таблиці з інших настільних і серверних баз даних для зберігання інформації, необхідної додатком. Приєднавши зовнішні таблиці, SQL Server користувач буде працювати з базами даних у цих таблицях так, як якщо б це були таблиці Access. При цьому і інші користувачі можуть продовжувати працювати з цими даними в тому середовищі, в якій вони були створені.

Наступним кроком доцільно проводити оцінку можливостей моделі щодо відповідності сформованим вимогам (задачам АСК). В разі позитивного результату такої оцінки відбувається перехід до побудови датологічної моделі, а в протилежному випадку - уточнення і доопрацювання інфологічної моделі.

В технології створення БД не менш важливим є дотримання наступних вимог:

- 1) оцінка інтенсивності транзакцій БД (реєстрації і читання документів). Дану оцінку доцільно проводити на основі експертної оцінки для кожного документу.[2]
- 2) режим доступності БД. Доступ користувачів до баз даних автоматизованої системи керування (АСК) здійснювати шляхом представлення сервісів, що дозволить формувати запити на отримання інформації обмеженого доступу, тобто із встановленою політикою прав доступу та вибіркового захисту, ранжируванням прав

доступу до БД, а також завданням розміру шифрованої інформації у файлі БД (наприклад, ідентифікацію і аутентифікацію користувачів проводити за індивідуальним паролем, за допомогою електронного ключа тощо) із урахуванням СКБД або шляхом вбудовування захисних механізмів в СКБД [7].

3) час відклику БД. Для транзакції читання не повинно перевищувати 2 секунди і 3 секунди для транзакції реєстрації (поповнення даних).

4) оцінка показників RTO і RPO БД. Максимальний час відновлення бази даних після серйозного збою (RTO) для БД визначити 2 години. Максимальний інтервал часу роботи, за який можуть бути втрачені зміни БД після відновлення (RPO) не повинен перевищувати 1 хвилини.

5) визначення рівня безпеки БД. Інформаційна безпека БД забезпечується не тільки засобами СКБД (сукупністю засобів керування створенням, використанням та обслуговуванням бази даних), але й вибором операційної системи. Операційну систему (ОС) в складі перспективної АСК доцільно мати у вигляді окремих версій для серверів та ПК (апаратне забезпечення серверів більш консервативне і має більший ресурс).

Основний акцент робити на засобах віртуалізації і засобах організації розподілених кластерних обчислювальних систем, які забезпечують створення кластерів різних типів (сховище, кластер високої доступності і кластер балансування навантаження) [7, 12].

Представлена інформаційна модель бази даних АСК надає можливість розробити програмне забезпечення АСК «Аеропорт», а її реалізація дозволить вирішувати відповідні функціональні задачі АСК підрозділами різних рівнів. Дослідження та розробка бази даних за ПЗ автоматизованої системи керування (АСК) аеропортом та в майбутньому можливості її застосування в сучасних аеропортах.

1.7 Концептуальне проектування

Концептуальне проектування БД - процедура (процес) створення інформаційної моделі підприємства, що не залежить від будь - яких фізичних умов реалізації.

Концептуальне проектування складається з 8 етапів:

1. Визначення типів сутностей
2. Визначення типів зв'язків
3. Визначення атрибутів і зв'язування їх з типами сутностей і зв'язків
4. Визначення доменів атрибутів
5. Визначення атрибутів, що є потенційними і первинними ключами
6. Спеціалізація або генералізація типів сутностей (необов'язковий етап)
7. Створення діаграми "сутність - зв'язок"
8. Обговорення з кінцевими користувачами.

Проектування бази даних полягає в побудові комплексу взаємозв'язаних моделей даних.

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

Розробка концептуальної моделі предметної області є найважливішим етапом проектування бази даних є, не орієнтованим на конкретну СКБД. Концептуальна модель предметної області будується першою та полягає у структуризації наочної області : об'єкти реального миру піддаються класифікації, фіксується сукупність тих, що підлягають відображенню в БД об'єктів. Для кожного об'єкту фіксується сукупність властивостей, за допомогою яких описуватимуться конкретні екземпляри об'єкту, і відносини (взаємозв'язки) з іншими об'єктами. Потім вирішуються питання про те, яка

інформація про об'єкти повинна бути представлена в БД і як її представити за допомогою даних.

1.8 Вибір моделі даних

Предметну область «Аеропорт» можна описати у вигляді моделі даних. Модель відображає реальний світ у деякі зрозумілі людині концепції, цілком незалежні від параметрів середовища зберігання даних. Існує безліч підходів до побудови таких моделей: графові моделі, семантичні мережі, модель «сутність-зв'язок» і т. д. Для опису досліджуваної предметної області будемо використовувати такі моделі даних як «сутність-зв'язок», реляційна модель та інші.

Сутність (entity) – це об'єкт, який може бути ідентифікований певним способом, що відрізняє його від інших об'єктів. В рамках даної предметної області виділені наступні сутності: Квитки, Клієнт, Країни, Літаки, Міста, Модель літаків, Посади, Рейси, Співробітники, Типи літаків,.

Сутність фактично являє собою безліч атрибутів, які описують властивості всіх членів даного набору сутностей. Наприклад, у сутності «Посада» були виділені наступні атрибути:

Посада (Код посадки, назва, оклад).

Набір зв'язків (relationship set) - це відношення між B (B -не менше 2) сутностями, кожна з яких відноситься до певного набору сутностей.

У випадку $B=2$, тобто коли зв'язок об'єднує дві сутності, вона називається бінарною. Доведено, що парний набір зв'язків ($B>2$) завжди можна замінити безліччю бінарних, однак перші краще відображають семантику предметної області.[3]

Число сутностей, яке може бути асоційоване через набір зв'язків з іншою сутністю, називають ступенем зв'язку. Розгляд ступенів особливо корисно для бінарних зв'язків. Можуть існувати наступні ступені бінарних зв'язків:

- 1) Один до одного (позначається 1 : 1). Це означає, що в такій зв'язку сутності з однією роллю завжди відповідає не більше однієї сутності з іншою роллю.
- 2) Один до багатьох (1 : Б). В даному випадку сутності з однією роллю може відповідати будь-яке число сутностей з іншою роллю.
- 3) Багато до одного (Б : 1). Ця зв'язок аналогічна відображенню 1:Б
- 4) Багато до багатьох (Б : Б). У цьому випадку кожна із асоційованих сутностей може бути представлена будь-яким кількістю примірників.

Модель «сутність-зв'язок» предметної області “Аеропорт” представлена на Рисунок.1.6

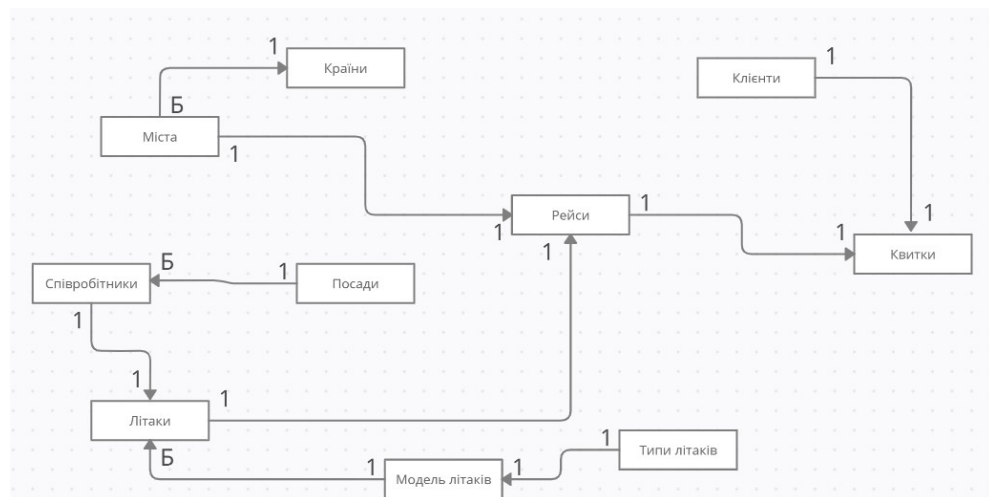


Рисунок 1.6- Діаграми «сутність - зв'язок»

1.7 Логічне проектування

Логічне проектування бази даних – це процес перетворення концептуальної моделі в логічну модель з урахуванням особливостей обраної СКБД.

Основним завданням логічного проектування є розробка логічної схеми, орієнтованої на вибрану СКБД. Оскільки переважна більшість сучасних СКБД - реляційні, то і концептуальну модель БД слід відображати на реляційну модель.[6]

У основі реляційної моделі використовується поняття “відносини”, яке використовується для уявлення набору екземплярів об'єкту (сутність) та відносин (зв'язків) між об'єктами. Відношення представляється як певним чином організована таблиця.

Для відображення інформаційної структури ПО на логічну схему реляційної БД слід визначити:

- 1) скільки таблиць і які, повинна включати БД;
- 2) які ступені (кількість стовпців) і склад кожної таблиці;
- 3) як встановлюються зв'язки між різними таблицями;
- 4) використання в різних таблицях одного і того ж ключа;
- 5) використання ключа однієї таблиці як атрибут (поля) в записі іншої таблиці (зовнішні ключі);
- 6) створення спеціальних таблиць, що пов'язують сутності згідно з ER-діаграмою;
- 7) як забезпечити повноту, несуперечність і узгодженість інформації, що зберігається в БД;
- 8) які атрибути (поля) використовуються як ключі;

Для зменшення надмірності інформації і виключення аномалій виконується нормалізація. Визначення змісту кожної таблиці зі зазначення типу даних та обмеження на значення для кожного атрибуту рекомендується навести у таблиці.

На підставі проведеного вище проектування необхідно:

- 1) при наявності складних зв'язків у ER – діаграмі (зв'язки типу Б:Б) перетворити їх на зв'язки типу 1:Б та 1:1, для чого ввести додаткові асоціативні сутності;

- 2) перетворити ER – діаграму у відношення, визначити необхідну кількість відношень, яка дорівнює кількості сутностей на ER – діаграмі;
- 3) для кожного відношення визначити всі атрибути;
- 4) для кожного атрибуту таблиці визначити вимоги до підтримки цілісності даних: визначити обов'язковість наявності даних (припустимість значення NULL);[7]
- 5) встановити обмеження для доменів атрибутів;
- 6) визначити тип даних для кожного атрибуту відношення;
- 7) результати аналізу навести у таблиці, яка створюється для кожній сутності.

1.8 Фізичне проектування

Згідно зі специфікацією по виконанню функцій підприємств "Аеропорт", було витягнуто усі іменники або їх поєднання з прикметниками для подальшого розподілу на сутності та їх атрибути. [4]

Вибрав найбільші об'єкти, а другорядні включив в них (в подальшому вони стануть атрибутами).

Задokumentувавши типи сутностей:

- 1) Присвоївши зрозумілі імена (Квитки, Клієнт, Країни, Літаки, Міста, Модель літаків, Посади, Рейси, Співробітники, Типи літаків.);
- 2) Визначив кількість екземплярів сутностей я з'ясував що їх десять, потрібно придумати їм певні атрибути.
- 3) Розмістивши всі дані в словник (таб.1.8)

Таблиця.1.8- Словник даних

Сутність	Атрибут	Тип даних MySQL
Квитки	Id квитка	Int
	Клієнт	Int
	Рейс	Int
	Ціна	Money
	Місце	Varchar(5)
	Заброньовано	Bit
Клієнт	Id клієнту	Int
	ПІБ	Varchar(50)
	Дата народження	date
	Паспорт	Varchar(50)
	Телефон	Varchar(12)
Країни	Id країни	Int
	Назва	Varchar(50)
Літаки	Id літака	Int
	Модель літака	Varchar(50)
	Дата випуску	Date
	Головний пілот	Int
Міста	Id міста	Int
	Назва	Varchar(50)
	Країна	Int
Модель літаків	Модель	Varchar(50)
	Пасажиромісткість	Int
	Тип	Int
	Вантажопідйомність (т)	Int
Посади	Id посади	Int
	Назва	Varchar(50)
	Оклад	money
Рейси	Id рейсу	Int
	Дата	Date

	Місто відльоту	Int
	Місто прильоту	Int
	Час відльоту	Time(7)
	Час прильоту	Time(7)
	Час у польоті	Time(7)
	Літак	Int
Співробітники	Id Співробітника	Int
	ПІБ	Varchar(50)
	Посада	Int
	Вік	Int
	Телефон	Varchar(12)
	Стать	Varchar(1)
Типи літаків	Id типу	int
	Назва	Varchar(50)

1.9 Висновок по першому розділу

В перший розділ присвячений опису предметної області що досліджується для написання випускної роботи. Розкривається актуальність данної теми та вибір предметної області для данного проєкту. Та ознайомлення з інфологічною моделлю предметної області для якої розробляється програма. Подано перелік поставлених задач для виконання поставленої мети випускної роботи, а саме Розробити програму з обліку попередніх та резервних заказів на прикладі Аеропорт та написати пояснювальну записку з повним оформленням.

РОЗДІЛ 2

РОЗРОБКА ОБ'ЄКТІВ ДЛЯ ДОСТУПА ДО РЕЛЯЦІЙНИХ ДАНИХ

2.1 Поняття реляційних модель даних

Реляційна модель даних — логічна модель даних. Вперше була запропонована британським ученим співробітником компанії IBM Едгаром Франком Коддом (E. F. Codd) в 1970 році в статті «A Relational Model of Data for Large Shared Data Banks». В даний час ця модель є фактичним стандартом, на який орієнтуються практично всі сучасні комерційні системи керування базами даних (СКБД).

У реляційній моделі досягається більш високий рівень абстракції даних, ніж в ієрархічній або мережевій. У згаданій статті Е. Ф. Кодда стверджується, що «реляційна модель надає засоби опису даних на основі тільки їх природної структури, тобто без потреби введення якоїсь додаткової структури для цілей машинного представлення». Іншими словами, подання даних не залежить від способу їх фізичної організації. Це забезпечується за рахунок використання математичного поняття відношення (сама назва «реляційна» походить від англійського relation — «відношення»).

До складу реляційної моделі даних зазвичай включають теорію нормалізації. Крістофер Дейт визначив три складові частини реляційної моделі даних:

- структурна
- маніпуляційна
- цілісна

Структурна частина моделі визначає, що єдиною структурою даних є нормалізоване парне відношення. Відношення зручно представляти у формі таблиць, де кожен рядок є кортеж, а кожен стовпець — атрибут, визначений на деякому домені. Даний неформальний підхід до поняття відношення дає більш звичну для розробників і користувачів форму представлення, де реляційна база даних являє собою кінцевий набір таблиць.

2.2 Маніпулювання даними

Маніпуляційна частина моделі визначає два фундаментальних механізми маніпулювання даними — реляційну алгебру і реляційне числення. Основною функцією маніпуляційної частини реляційної моделі є забезпечення заходів реляційності будь-якої конкретної мови реляційних БД: мова називається реляційною, якщо вона має не меншу виразність і потужність, ніж реляційна алгебра або реляційне числення.

Цілісна частина моделі визначає вимоги цілісності сутностей і цілісності посилань. Перша вимога полягає в тому, що будь-який кортеж будь-якого відношення відмінний від будь-якого іншого кортежу цього відношення, тобто іншими словами, будь-яке відношення має володіти первинним ключем. Вимога цілісності щодо посилань, або вимога зовнішнього ключа полягає в тому, що для кожного значення зовнішнього ключа, що з'являється у відношенні, на яке веде посилання, повинен знайтися кортеж з таким же значенням первинного ключа, або значення зовнішнього ключа повинно бути невизначеним (тобто ні на що не вказувати).

Можна провести аналогію між елементами реляційної моделі даних і елементами моделі «сутність-зв'язок». Реляційні відносини відповідають наборам сутностей, а кортежі — сутностям. Тому, як і в моделі «сутність-зв'язок», стовпці в таблиці, що представляє реляційне відношення, називають атрибутами.

Кожен атрибут визначений на домені, тому домен можна розглядати як множина допустимих значень даного атрибуту. Кілька атрибутів одних відношень і навіть атрибути різних відношень можуть бути визначені на одному і тому ж домені.

Іменована множина пар «ім'я атрибута — ім'я домену» називається схемою відношення. Потужність цієї множини — називають ступенем чи «парністю» відносини. Набір іменованих схем відносин являє собою схему бази даних.

Таблиця. 2.1 - Словник даних, та використані ключі

Сутність	Атрибут	Тип даних MySQL	Ключі
Квитки	Id квитка	Int	Primary Key
	Клієнт	Int	Foreign Key
	Рейс	Int	Foreign Key
	Ціна	Money	
	Місце	Varchar(5)	
	Заброньовано	Bit	
Клієнт	Id клієнту	Int	Primary Key
	ПІБ	Varchar(50)	
	Дата народження	date	
	Паспорт	Varchar(50)	
	Телефон	Varchar(12)	
Країни	Id країни	Int	Primary Key
	Назва	Varchar(50)	
Літаки	Id літака	Int	Primary Key

Таблиця. 2.1 - Словник даних, та використані ключі

	Модель літака	Varchar(50)	
	Дата випуску	Date	
	Головний пілот	Int	
Міста	Id міста	Int	Primary Key
	Назва	Varchar(50)	
	Країна	Int	Foreign Key
Модель літаків	Модель	Varchar(50)	Primary Key
	Пасажиромісткість	int	
	Тип	Int	Foreign Key

	Вантажопідйомність (т)	Int	
Посади	Id посади	Int	Primary Key
	Назва	Varchar(50)	
	Оклад	money	
Рейси	Id рейсу	Int	Primary Key
	Дата	Date	
	Місто відльоту	Int	Foreign Key
	Місто прильоту	Int	Foreign Key
	Час відльоту	Time(7)	
	Час прильоту	Time(7)	
	Час у польоті	Time(7)	
	Літак	Int	Foreign Key
Співробітники	Id Співробітника	Int	Primary Key
	ПІБ	Varchar(50)	

Таблиця. 2.1 - Словник даних, та використані ключі

	Посада	Int	Foreign Key
	Вік	Int	
	Телефон	Varchar(12)	
	Стать	Varchar(1)	
Типи літаків	Id типу	Int	Primary Key
	Назва	Varchar(50)	

Атрибут, значення якого однозначно ідентифікує кортежі, називається ключовим (або просто ключем). Якщо кортежі ідентифікуються тільки зчепленням значень декількох атрибутів, то говорять, що відношення має складовий ключ. Ставлення може містити кілька ключів. Завжди один із ключів оголошується первинним, його значення не можуть оновлюватися. Всі інші ключі відносини називаються можливими ключами.

На відміну від ієрархічної і мережної моделей даних в реляційній відсутнє поняття групових відношень. Для відображення асоціацій між кортежами різних відносин використовується дублювання їх ключів.

Переваги реляційної моделі:

- простота і доступність для розуміння користувачем. Єдиною використовуваною інформаційною конструкцією є «таблиця»;
- суворі правила проектування, які базуються на математичному апараті;
- повна незалежність даних. Зміни в прикладній програмі при зміні реляційної БД мінімальні;
- для організації запитів і написання прикладного ПЗ немає необхідності знати конкретну організацію БД у зовнішній пам'яті.

Недоліки реляційної моделі:

- далеко не завжди предметна область може бути представлена у вигляді «таблиць»;
- в результаті логічного проектування з'являється множина «таблиць». Це призводить до труднощів розуміння структури даних;
- БД займає відносно багато зовнішньої пам'яті;
- відносно низька швидкість доступу до даних.

2.3 Висновок по другому розділу

В другому розділу присвячений розробки доступу до бази даних. Вибір та опис реляційної моделі та використання в даному проєкті.

РОЗДІЛ 3

ПРОЕКТУВАННЯ БАЗИ ДАНИХ

3.1. Апаратні вимоги до програмно забезпечення

Мінімальні апаратні вимоги до мого ПЗ :

Операційна система не нижче Windows 7

Процесор Pentium з мінімальною тактовою частотою 2 Ghz

Оперативна пам'ять - не менше 2 GB

Необхідно 0,5 ГБ вільного простору (Не враховуючи обсяг програм, що займають біля ~ 20 GB)

Кольоровий графічний дисплей (роздільна здатність 1024 x 768)

Маніпулятор типу миша і клавіатура

Наявність бд, та ставновленого MySQL Server

Рекомендовані системні вимоги до мого ПЗ:

Операційна система: Windows 10

Процесор: Dual core x64 2.5 Ghz

Необхідні програми: SQL Management Studio 2017, .NET 4.-3---4.-7

Оперативна пам'ять: 4 ГБ

Необхідно 0,5 ГБ вільного простору (Не враховуючи обсяг програм, що займають біля ~ 20 GB)

Наявність бд, та ставновленого MySQL Server

3.2. Створення входу на сервер на базі фіксованих серверних ролей

Створення бази даних проводиться за допомогою команди мови Transact SQL - Create database. Код для створення бази даних «Аеропорту» представлено нижче:

Лістинг коду 3.1 – Створення бази даних

```
CREATE DATABASE airport ON PRIMARY(
```

```
NAME = 'airport',
```

```
FILENAME = 'C:\database\ 'Airport.mdf',
```

```
SIZE = 5MB,
```

```
FILEGROWTH = 20%)
```

```
LOG ON(
```

```
NAME = N' Airport _log',
```

```
FILENAME = N'C:\database\ 'Airport_log.ldf',
```

```
SIZE = 5MB,
```

```
FILEGROWTH = 20%)
```

На далі треба створити таблиці завдяки нашим атрибутам та сутностям які ми зробили раніше.

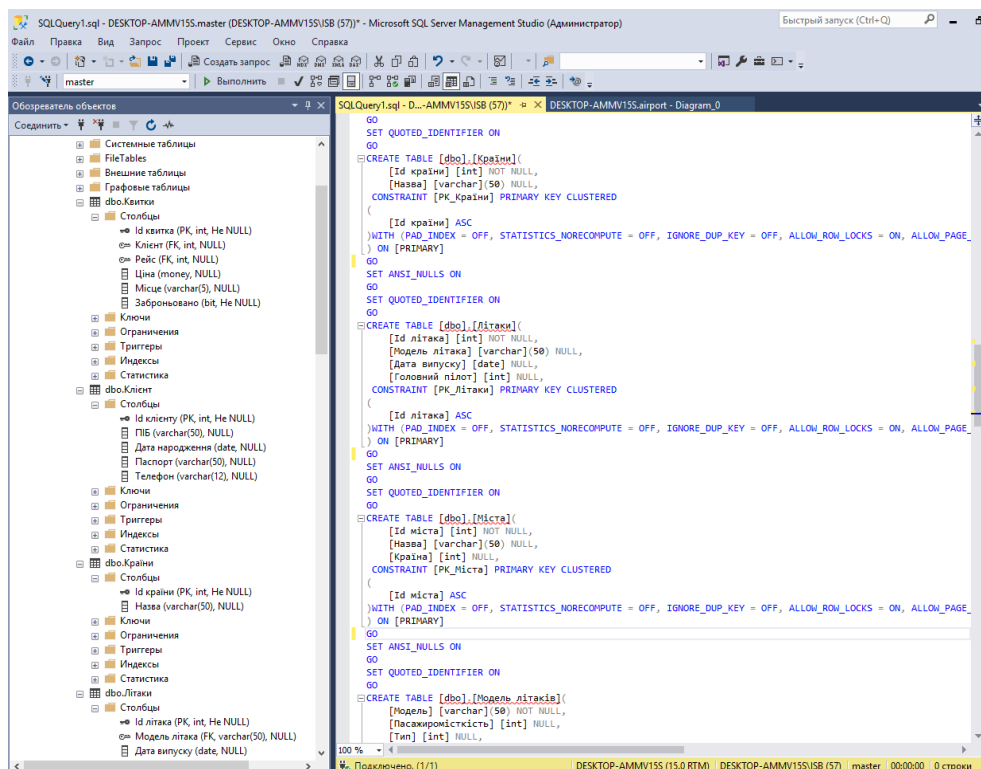


Рисунок. 3.1 – Створення таблиць

Виходячи з нашого технічного плану треба відтворити систему захисту бази даних

У SQL Server 2017 є зумовлені серверні ролі, що володіють різними правами. Їх зміна і видалення заборонено. Кожна роль має свої права. Щоб надати ці права користувачеві, потрібно додати його обліковий запис в склад ролі сервера. Знаходження імені входу в ролі public (присутній завжди) не дає можливості на перегляд об'єктів СКБД. Виходячи з концепції безпеки наведеної вище, створимо такі імена входів і наділимо їх відповідними правами відповідними правами.

У навчальних цілях імена входу створюються без пароля. У зв'язку з цим додатковими опціями були відключені політика безпеки паролів. Для забезпечення безпеки вбудоване ім'я входу «sa» було відключено. Лістинг команд, для створення імен входу наведено нижче.

Лістинг 3.2 - Створення імені «Admin»

```
USE airport
CREATE LOGIN Admin
WITH PASSWORD=N'',
DEFAULT_DATABASE=airport,
CHECK_EXPIRATION=OFF,
CHECK_POLICY=OFF
```

Далі необхідно внести імена входу в серверні ролі. Лістинг команд наведено нижче.

Внесення імені входу «Administrator» в роль «sysadmin».

```
USE airport
EXEC master..sp_addsrvrolemember
@loginame = N'Admin',
@rolename = N'sysadmin'
```

3.3. Створення ролей рівня баз даних

Для розмежування доступу на рівні об'єктів бази даних потрібно створити ролі бази даних, а потім внести в них користувачів бази даних.

Щоб створити власні ролі рівня бази даних застосовують наступний синтаксис:

```
sp_addrole [@rolename =] 'роль'.
```

У цій роботі були створені такі ролі:

- Mydeny: дана роль була створена для повної заборони на системні таблиці, уявлення і процедур, а також на прямий доступ до призначених для користувача таблиць. Вона також забороняє створювати об'єкти в БД.

- Administrator (Admin) він володіє абсолютними правами.

Лістинг команд створення ролі бази даних «Mydeny» і надання їй прав наведено нижче.

Створення ролі «Mydeny» в базі даних «Аеропорт».

```
USE airport
```

```
CREATE ROLE Mydeny
```

```
EXEC sp_addrolemember N'db_denydatawriter', N'Mydeny'
```

3.4. Створення користувачів баз даних

Користувачі бази даних призначені для розподілу прав доступу до об'єктів на рівні бази даних. Для кожного імені входу повинні бути створені користувачі в базі даних «Аеропорт», і включені до відповідних ролей.

Лістинг команд створення користувачів баз даних і зв'язку їх з іменами входу наведено нижче.

Створення користувачів бази даних «Аеропорт» і зв'язування їх з іменами входу.

```
USE airport
```

```
CREATE USER Admin FOR LOGIN Admin
```

Потім користувачів баз даних потрібно включити до відповідних ролей. Лістинг команд включення користувачів бази даних в ролі наведено нижче. Включення користувачів баз даних в ролі для бд «Аеропорт».

```
USE master
```

```
EXEC sp_addrolemember N'Mydeny', N'airport'
```

3.5. Створення завдань резервного копіювання та операторів

Як правило, перед створенням завдань резервного копіювання створюють пристрої резервного копіювання. Це віртуальні пристрої, іменовані, які посилаються на файл або пристрій резервної копії. У випадку з магазином можна обійтися одним пристроєм резервного копіювання.

Створення пристрою «Dev»

```
EXEC airport.dbo.sp_addumpdevice
```

```
@devtype = N'disk',
```

```
@logicalname = N'Dev',
```

```
@physicalname = N'C:\Airport\MainDev.bak'
```

Для віддаленого відстеження виконання операцій обслуговування СКБД в MS SQL Server 2008 передбачені оператори. За допомогою них можна передавати звіти про виконання робіт обслуговування СКБД і про виниклі проблеми.

Лістинг команд, для створення оператора наведено нижче.

```
USE airport
```

```
EXEC msdb.dbo.sp_add_operator @name=N'Admin',
    @enabled=1,
    @weekday_pager_start_time=90000,
    @weekday_pager_end_time=180000,
    @saturday_pager_start_time=90000,
    @saturday_pager_end_time=180000,
    @sunday_pager_start_time=90000,
    @sunday_pager_end_time=180000,
    @email_address=N'admin@gmail.com',
    @category_name=N'[Uncategorized]'
```

3.6. Використання криптографічних алгоритмів для захисту

Можливість використовувати вбудовані засоби шифрування з'явилася, починаючи з 2005 версії продукту. У попередніх версіях для шифрування даних доводилося використовувати рішення сторонніх виробників. У версіях SQL Server 2005 \ 2017 рік є можливість виконання наступних криптографічних операцій:

- Створення сертифікатів;
- Створення асиметричних ключів;
- Створення симетричних ключів;
- Хешування даних;
- Шифрування і розшифрування інформації.

У загальному випадку робота з криптографічними інструментами СКБД починається з створення сертифікатів і ключів, потім може бути виконані операції шифрування і розшифрування інформації і хешування.

Створення ключів зазвичай виконує Admin баз даних або відповідальна особа (начальник), що має відповідні права. [8].

Команди безпосередньої обробки інформації виконуються користувачами .

Лістинг команд для створення криптографічного захисту інформації в базі даних:

Майстер ключ для бази даних «Аеропорт»:

```
Create master key encryption by Password = '122334'
```

Сертифікат для користувача Admin і призначення його власником сертифікату.

Час дії - починаючи з сьогоднішньої дати протягом двох років.

```
Create Certificate Admin _1 Authorization Admin
```

```
With Subject = 'Ключ співробітника 1',
```

```
Start_Date = '21/03/2022', Expiry_Date = '21/07/2022'
```

Симетричний ключ для користувача Admin, зашифрований за допомогою сертифіката Admin 1.

```
Create Symmetric Key Admin _key
```

```
With Algorithm = TRIPLE_DES
```

```
Encryption By Certificate Admin _1;
```

Надання користувачеві Admin прав для роботи з ключем Admin_key

```
Grant Control On Symmetric Key:: Admin_key to Admin
```

Реєстрація під логіном Admin

```
Open Symmetric Key Admin _key
```

```
decryption by certificate Admin_1;
```

Шифрування даних, що мають відношення до користувача Admin в таблиці постачальником

```
Select * from Літаки
```

```
go
```

```

Update dbo.Літаки
Set Passport = EncryptByKey(Key_GUID('Admin_key  '), Passport, 1,
Convert( varbinary, [Id Літака]))
where [Id Літака] =1;
go
Select * from Літаки

```

Для розшифрування даних слід виконати команду:

```

Update dbo.Літаки Set Passport = DecryptByKey(Passport, 1, Convert( varbinary, [Id
Літака])) where [Id Літака] =1;

```

В результаті будуть розшифровані тільки записи, значення коду літака в яких дорівнює 1.

Після завершення роботи з інформацією слід закрити ключ:

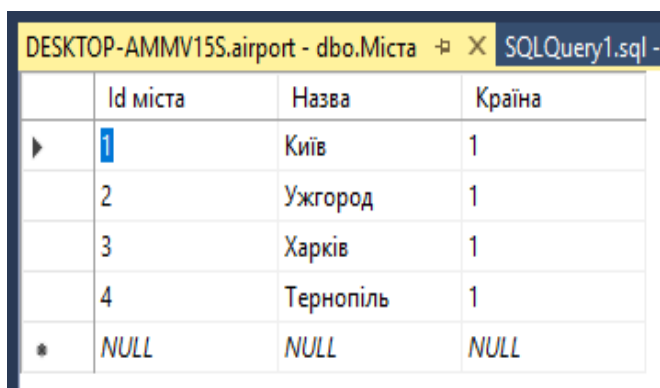
```
Close Symmetric Key Администратор_key
```

3.7 Фізична реалізація та тестування проекту

Після створення схеми даних, заповнюються таблиці. Спочатку заповнюються ті таблиці в яких зв'язок (1:Б), а потім (Б:1) .

	Id країни	Назва
▶	1	Україна
	2	США
*	NULL	NULL

Рисунок.3.2 – Заповнення таблиці «Країни»



	Id міста	Назва	Країна
▶	1	Київ	1
	2	Ужгород	1
	3	Харків	1
	4	Тернопіль	1
*	NULL	NULL	NULL

Рисунок.3.3 – Заповнення таблиці «Міста»

Заповнення можна проводити завдяки інтерфесу програми , або через скрипти SQL, або заповнити через інтерфейс котрий також створений на мові С# та показан на рисунках нижче. У майбутньому потрібно буде розвинути та додати до програми функцію виконання, та перегляд запитів, на даний час, запити можливо виконати в MySQL Console або ManagementStudio.

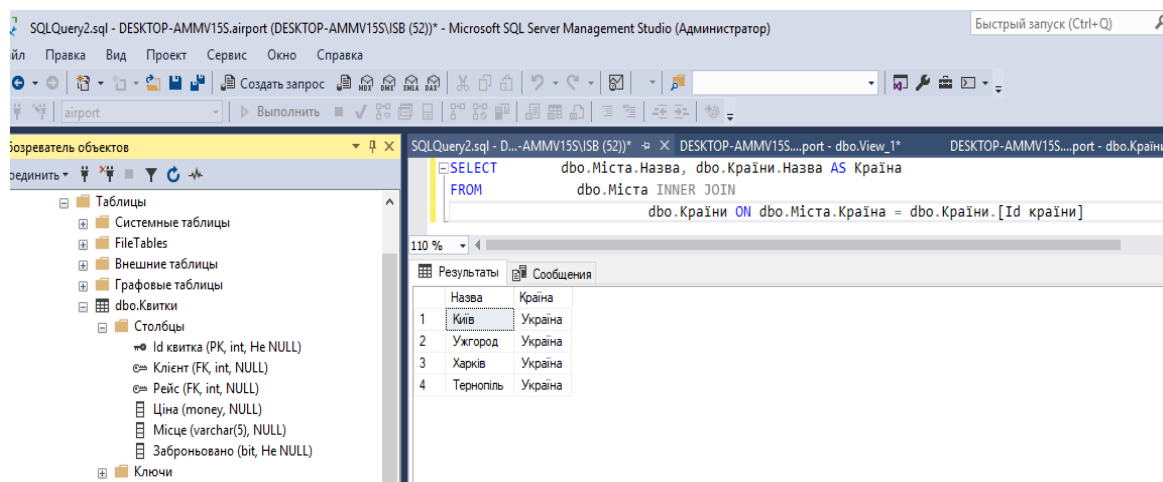


Рисунок.3.4 – Запит до створених таблиць

Надалі треба створити інтерфейс таблиць в режимі Visual Studio 2017 (С#).Почнемо з підключення бд до програмного середовища створення вікна авторизації.

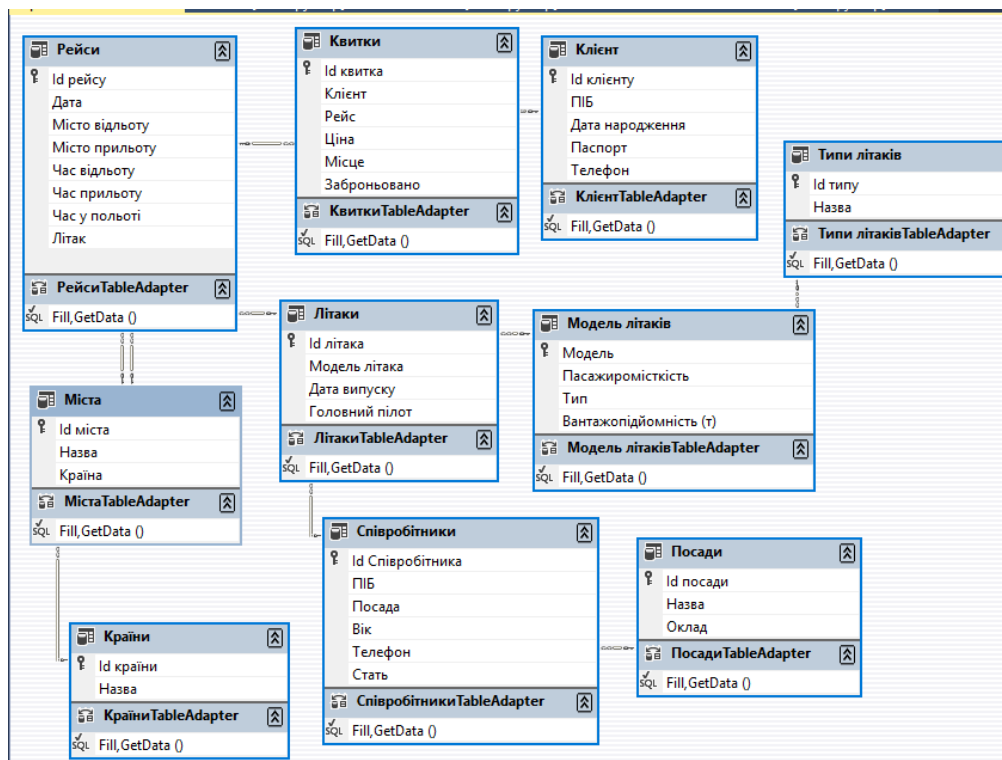


Рисунок.3.5 – Запит до створених таблиць

```

16     public Form1()
17     {
18         InitializeComponent();
19     }
20
21     private void button1_Click(object sender, EventArgs e)
22     {
23         DialogResult dialog = MessageBox.Show(
24             "Ви дійсно хочете завершити програму ? ",
25             "Вихід",
26             MessageBoxButtons.YesNo,
27             MessageBoxIcon.Warning
28         );
29         if (dialog == DialogResult.Yes)
30         {
31             Application.Exit();
32         }
33     }
34
35     private void button7_Click(object sender, EventArgs e)
36     {
37         if (textBox1.Text == "Admin")
38         {
39             if (textBox2.Text == "122334")
40             {
41                 this.Hide();
42                 MessageBox.Show("Авторизація успішна!");
43                 Form2 f2 = new Form2();
44                 f2.ShowDialog();
45             }
46         }
47         else
48         {
49             MessageBox.Show("Неправильний логін або пароль!!!");
50             textBox1.Clear();
51             textBox2.Clear();
52         }
53     }

```

Рисунок.3.6 – Код авторизації

Створюю форми у моєму випадку їх буде тринадцять, підключаємо їх до необхідних функцій, а також, додаємо кнопки до яких підключаємо скрипти оновлення, виходу, та ін.

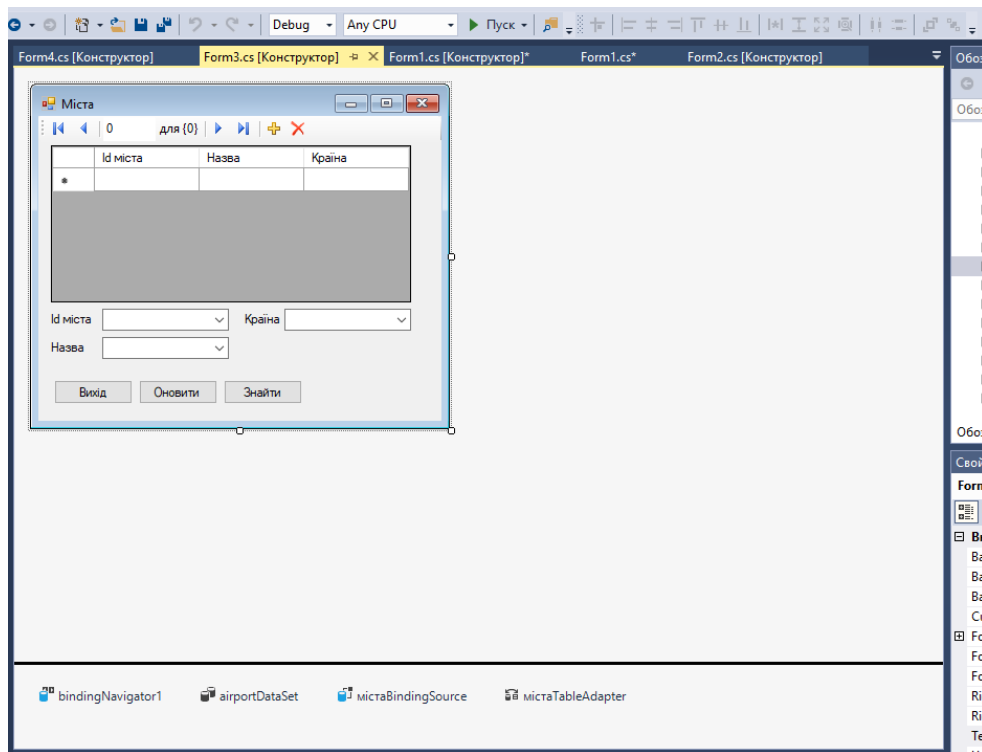


Рисунок.3.7 – Створення форми

З написаних функцій хотів би зупинись на кнопці оновлення котра перевіряє дані таблиці чи було щось змінено, додано, чи видалено, та після чого приймає зміни та заносить й зберігає їх у БД.

```
airportDataSet.MicraDataTable deletedTable = (airportDataSet.MicraDataTable)
    airportDataSet.Micra.GetChanges(DataRowState.Deleted);

airportDataSet.MicraDataTable newTable =
(airportDataSet.MicraDataTable)
    airportDataSet.Micra.GetChanges(DataRowState.Added);
```

```
airportDataSet.MicraDataTable modifiedTable =  
(airportDataSet.MicraDataTable)  
airportDataSet.Micra.GetChanges(DataRowState.Modified);
```

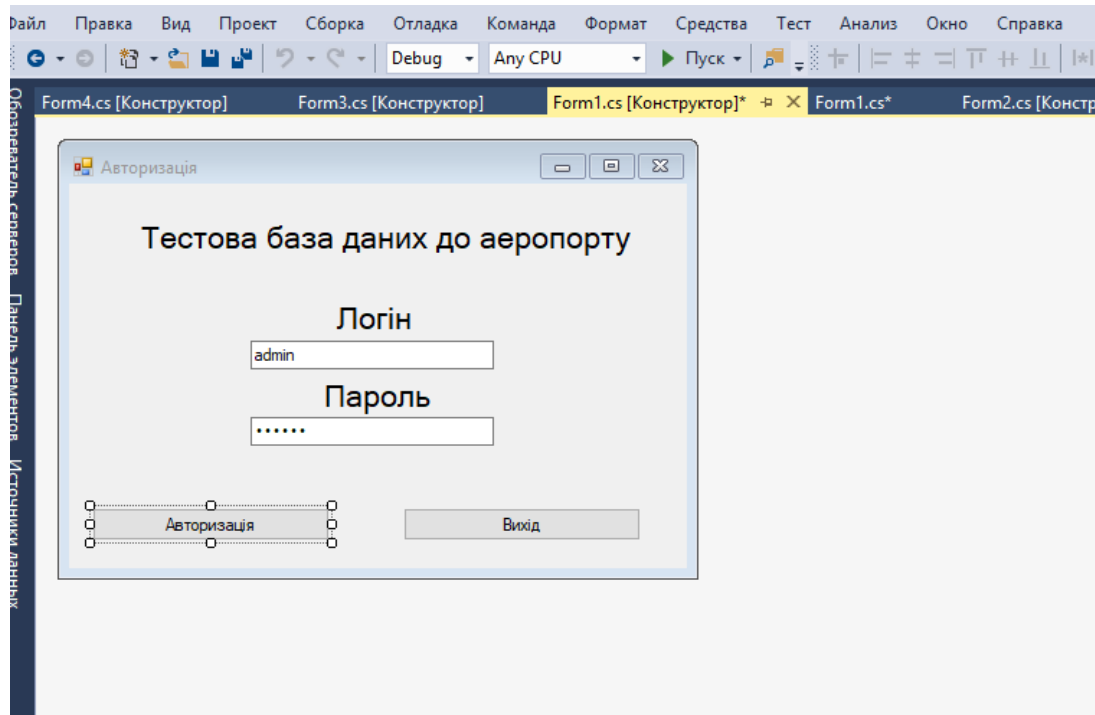


Рисунок.3.8 – Вікно авторизація

Проект, та увесь код буде у додатку. Після створення усіх необхідних форм , потрібно провести тестування ПЗ. Якщо в окні авторизації вказати не вірні дані , програма скаже що дані не вірні.

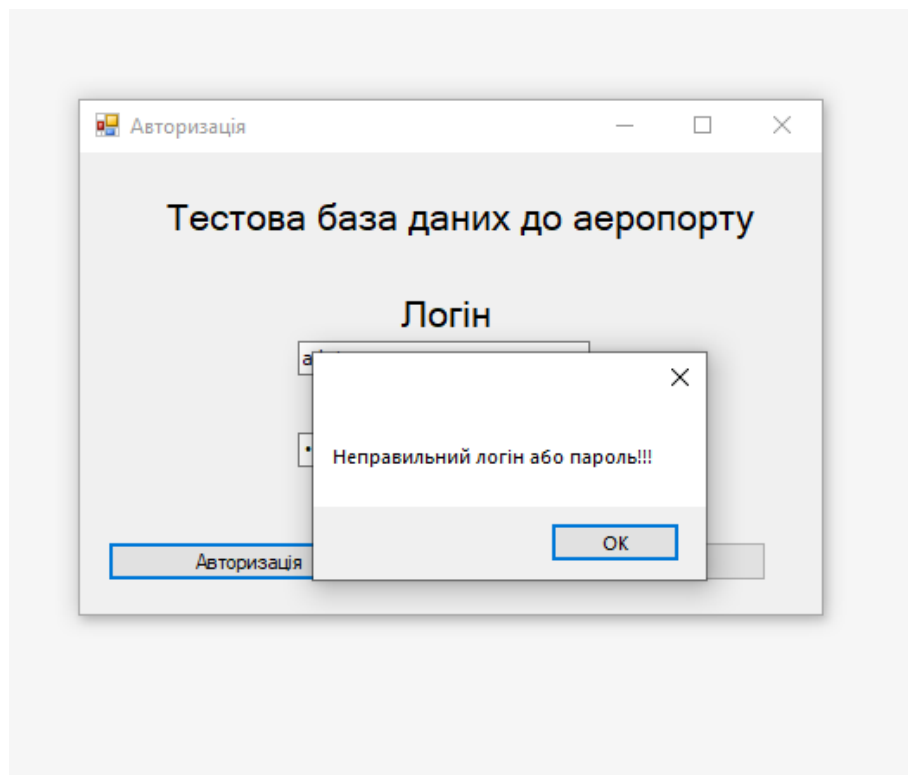


Рисунок.3.9 – Тестування авторизації

В програмі також є захист від помилкового натискання на кнопку «Вийти»

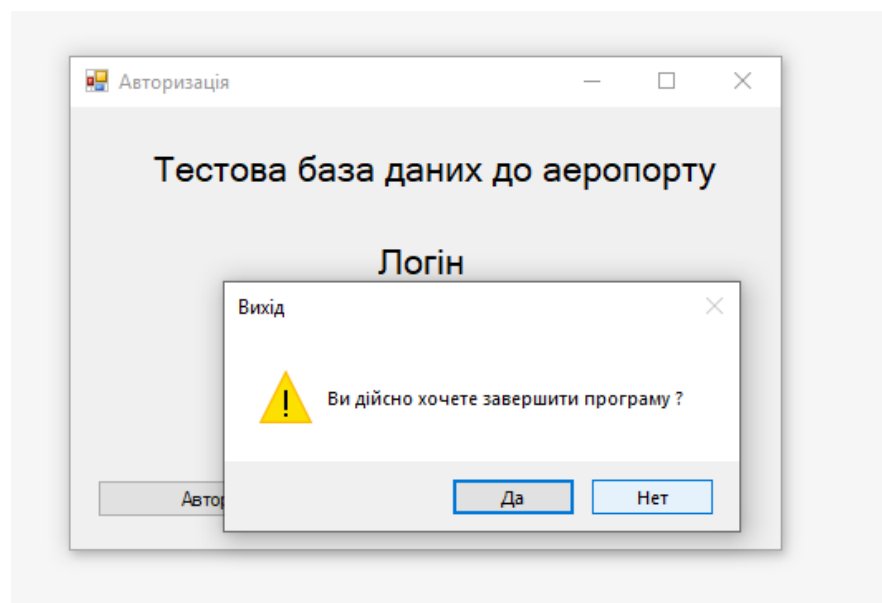


Рисунок.3.10 – Захист від помилкового натискання на кнопку «Вийти»

Після авторизації , відкриється меню програми , яке надає змогу обрати необхідну таблицю.

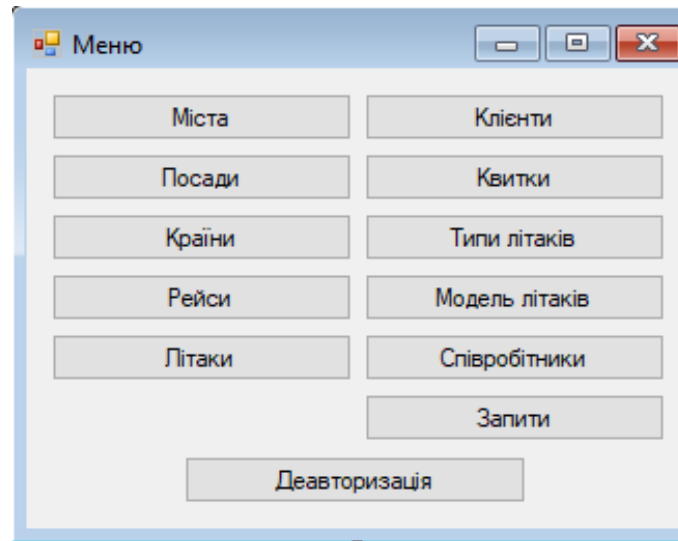


Рисунок.3.11 – Меню

Обравши таблицю «Міста» натискаємо на відповідно кнопку , та попадаємо у відповідне вікно

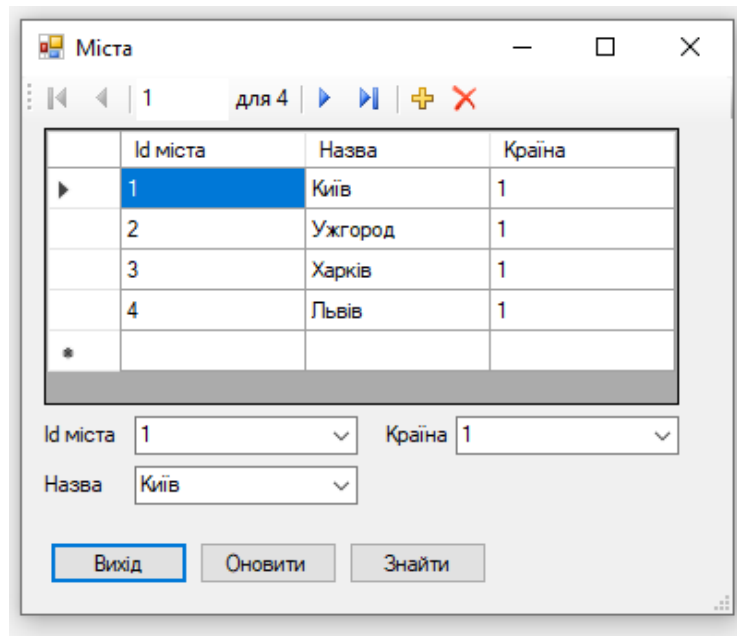


Рисунок.3.12 – Таблиця «Міста»

В програмі є кнопка оновити , вона створена для оновлення даних у базі даних (якщо користувач вніс зміни до якихось даних , він повинен натиснути кнопку оновити

, аби дані підтвердились), та їх зміна збереглась це може бути «редагування , видалення чи додання» нових даних до таблиць. Коли користувач вносить якісь зміни, у лівій стороні таблиці з'являється значок олівця , що каже про те , що вносяться зміни , та вони не підтвержені.

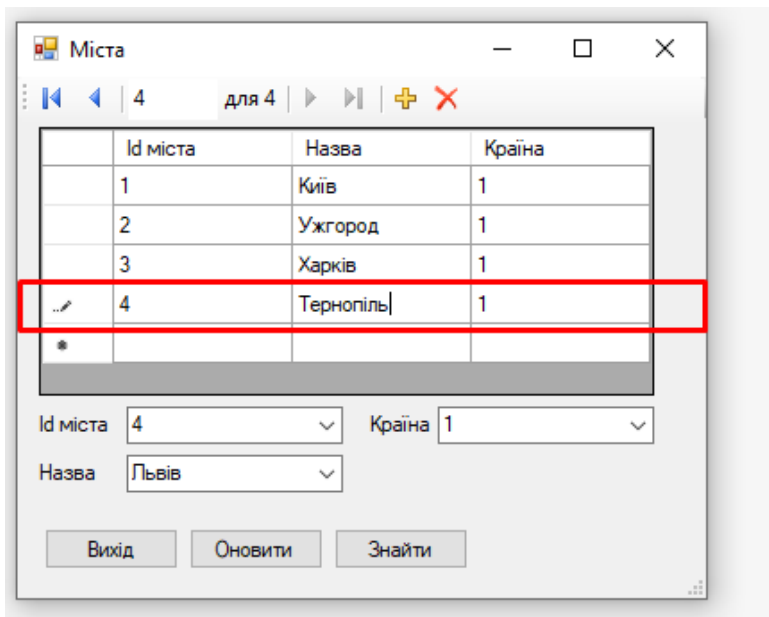


Рисунок.3.13 – Редагування даних

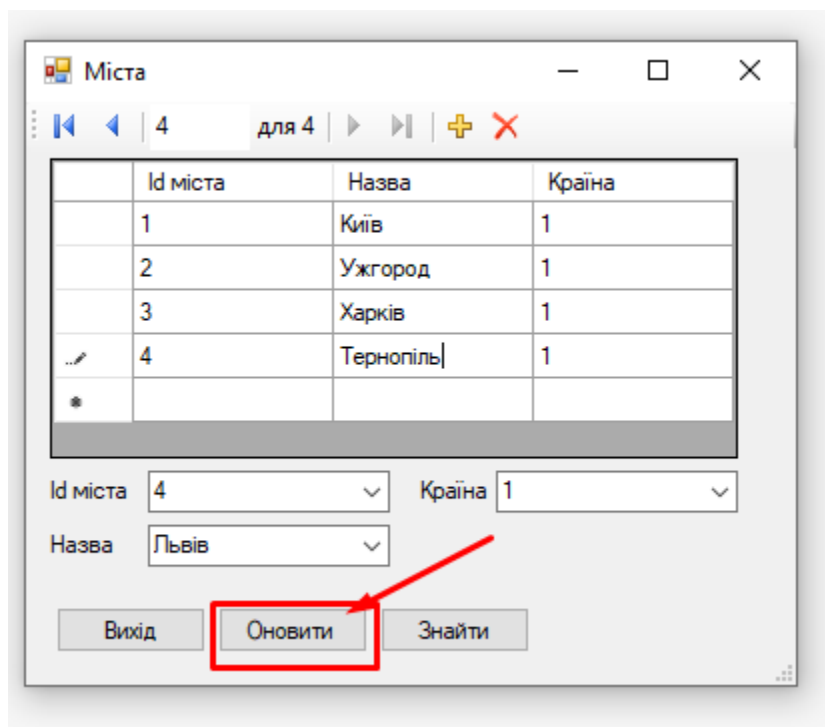


Рисунок.3.14 – Оновлення даних

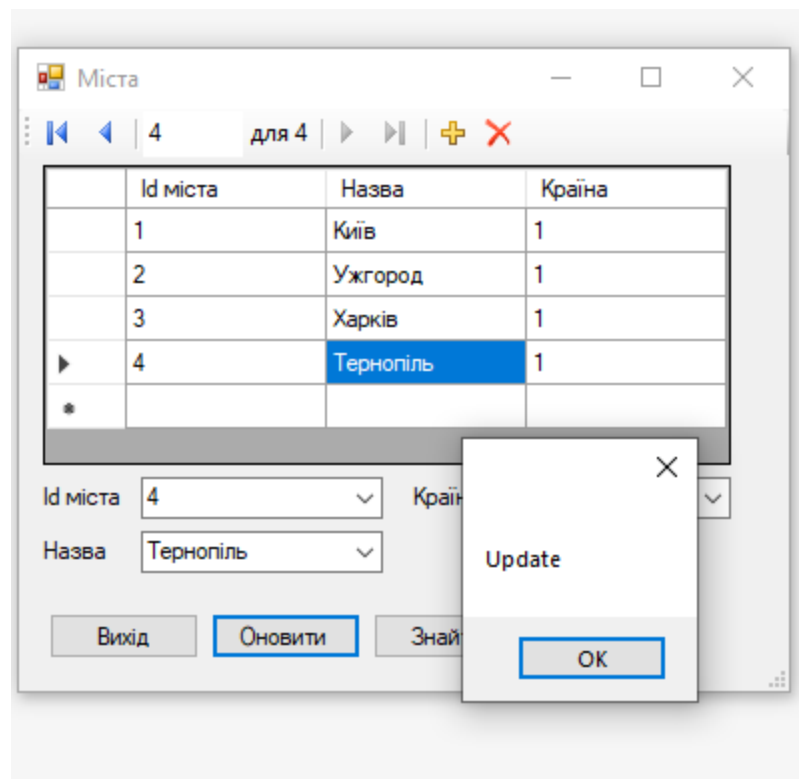


Рисунок.3.15 – Оновленні дані

Також у вікні таблиці «Міста» є так звана «пошукова система» вона може зробити пошук по всій таблиці та знайти необхідні дані . Вона створена для легкої роботи з великою кількістю даних.

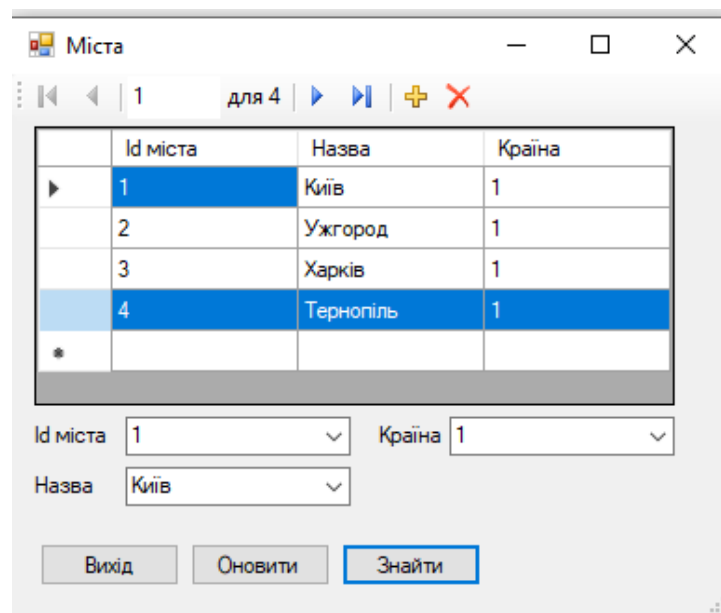


Рисунок.3.16 – Коректний відгук пошукової системи

У поле яке зображено на рис 4.16 потрібно ввести Id , чи назву запиту з пошуку , після чого натиснути «Пошук» необхідна строка таблиці буде виділена.

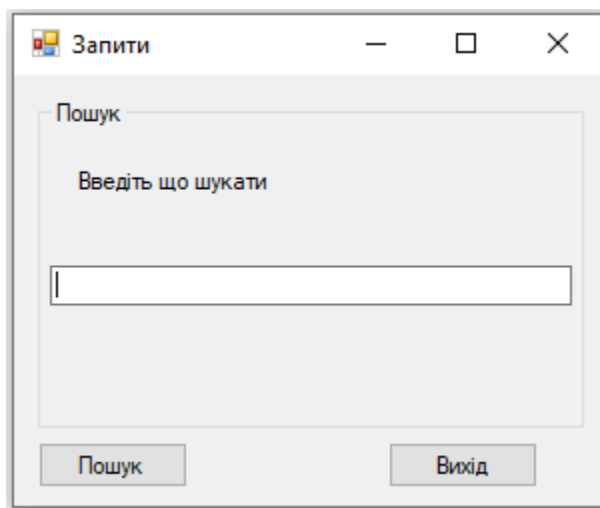


Рисунок.3.17 – Вікно вводу запиту для пошуку

Для тесту я знаходим строку в котрій назва міста була «Тернопіль»

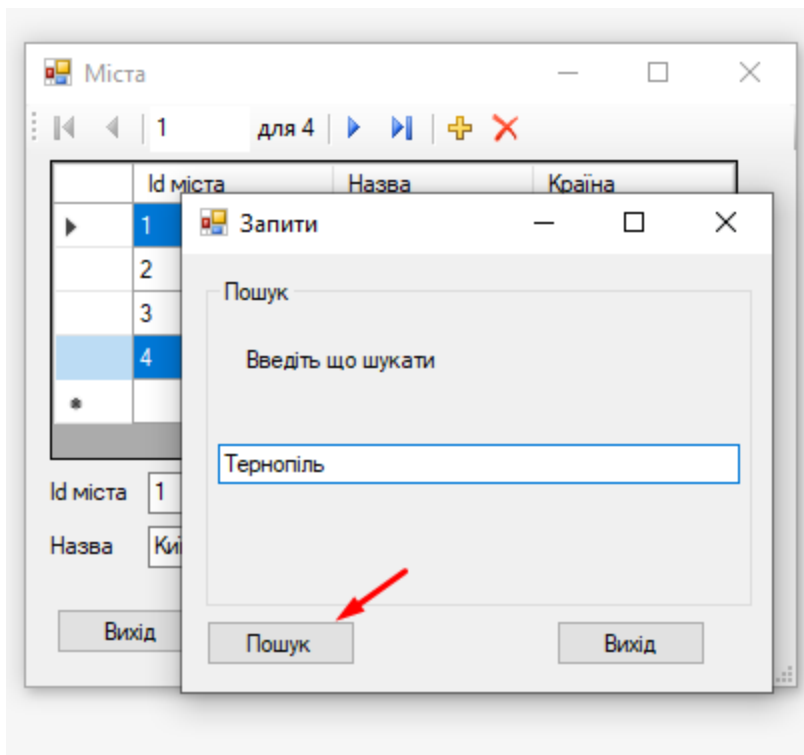


Рисунок.3.18 – Тестування пошукової системи

У програмі також є ще одне меню, котре можна буде розширювати, завдяки створеним представленням, створивши які, можна використовувати запити, та розшифрувати інші ключові значення котрі є у інших таблицях.

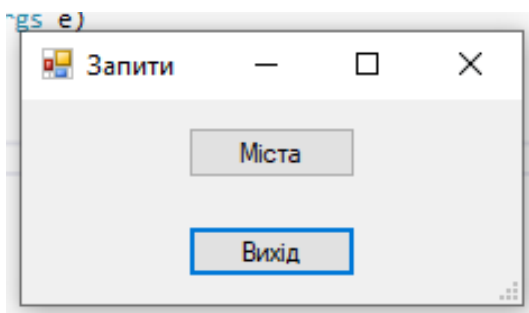


Рисунок.3.19 – Під меню запитів

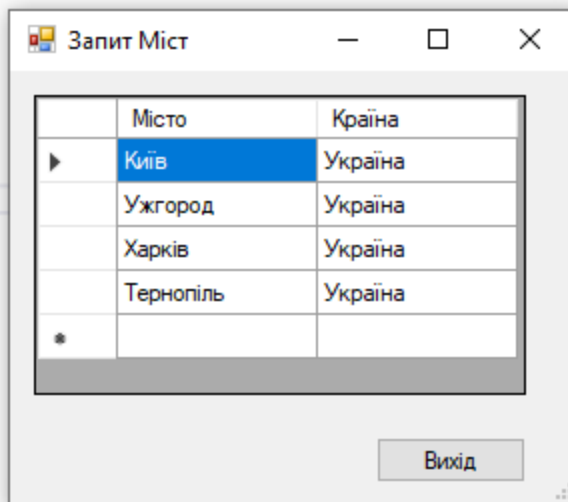


Рисунок.3.20 – Форма запиту «Міста»

У таблиці «Міста» додамо тестове місто Нью-Йорк из країни США, та натиснемо кнопку «Оновити».

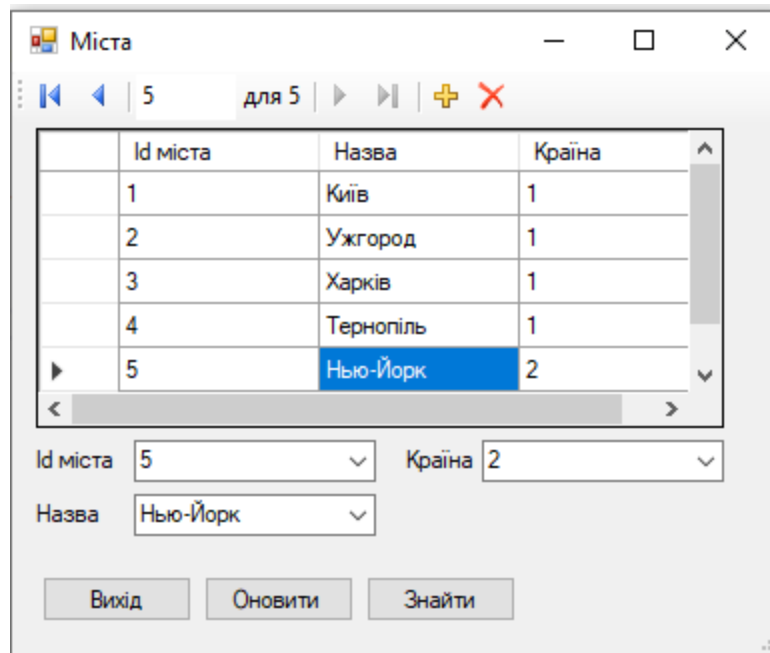


Рисунок.3.21 – Таблиця «Міста»

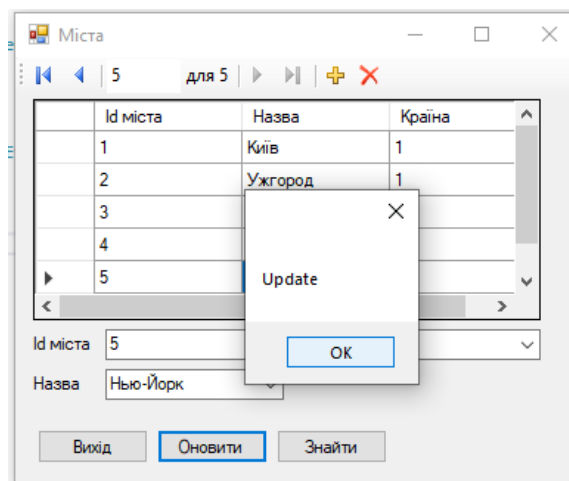


Рисунок.3.22 – Таблиця «Міста»

Після чого потрібно, знову перейти в «Запити» - «Міста», щоб побачити, що все працює коректно.

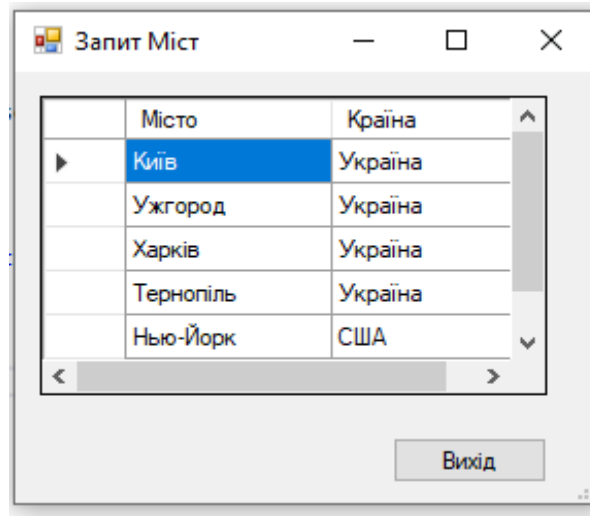


Рисунок.3.23 – Таблиця «Міста»

3.8 Висновок по третьому розділу

В третьому розділі було розроблено комплекс програм по швидкому пошуку інформації та обліку попередніх та резервних заказів, який відповідає основним вимогам до сучасних програмних розробок, для забезпечення автоматизації роботи аеропорту: точність розрахунків, можливість оновлення квартир, зручний інтерфейс, мінімально спрощений для користувача

ВИСНОВКИ

В даній дипломній роботі було проведено аналіз методів автоматизації обробки облікової інформації і, як результат, було розроблено комплекс програм по швидкому пошуку інформації та обліку попередніх та резервних заказів, який відповідає основним вимогам до сучасних програмних розробок, для забезпечення автоматизації роботи аеропорту: точність розрахунків, можливість оновлення квартир, зручний інтерфейс, мінімально спрощений для користувача. Це дає нам право говорити про конкурентоспроможність розробленої програми і можливість практичного її використання для вирішення реальних завдань.

На прикладі програми «Аеропорту» в якій можливо переглядати та редагувати дані аеропорту. Ми познайомилися з інструментом розробки баз даних Microsoft SQL Server . З їх допомогою можна швидко створювати бази даних для різних сфер діяльності людини.

Програмна система «Аеропорту» дозволяє оперативно переглядати, клієнтів, список рейсів, моделі літаків, типи літаків, список співробітників та багато іншого.

Закріпити знання, одержані під час вивчення дисципліни, а також засвоїти етапи розробки реляційних баз даних, прийоми використання засобів підтримки цілісності даних у базі, прийоми створення додатків для роботи з реляційною базою даних, правила оформлення документів на програмні вироби.

Для створення своєї бази даних необхідно мати хоча б один ПК із встановленим в ньому необхідним для цього пакетом програм. У процесі даної дипломної роботи була спроектована та реалізована в СКБД MySQL інформаційна система «Аеропорт».

У цій базі даних зберігається довідкова інформація, що міститься у певних главах посібника з прикладного програмування (Стефан Фаро, Рефакторинг SQL-приложений). База містить запити, що дозволяють здійснювати пошук необхідних даних та відображати статистичну інформацію, як то: інформацію про літаки, співробітників, моделі літаків, рейси, міста, та багато іншого.

Дана система пройшла всі три етапи проектування. На інфологічному рівні структура бази даних була відображена у вигляді ER-діаграми, яка надалі була приведена до третьої нормальної форми. На даталогічному рівні — представлена реляційною моделлю. У таблицях був усунений надлишок. Безпосередня робота з СКБД з формування таблиць і їх заповнення на комп'ютері була проведена на стадії фізичного проектування.

Таким чином, було створено 10 таблиць, та 15 форм. Надалі дану систему можна вдосконалювати, відповідно до потреб користувачів.

Впровадження даної розробки дозволило:

- скоротити час на придбання квитків;
- підвищити якість і точність розрахунків;
- скоротити час на складання і перевірку звітної документації;
- скоротити час на пошук необхідної інформації;

Всі використані в роботі матеріали і концепції з опублікованої наукової літератури та інших джерел мають посилання на них в списку літератури.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. [Електронний ресурс] b.ru/referaty_po_medicine/sozдание_bazy_dannyx_v.html
2. Обґрунтування вибору субд [Електронний ресурс] Режим доступу: [www.URL: https://studfile.net/preview/5093111/page:4/](http://www.studfile.net/preview/5093111/page:4/) (дата звернення:15.06.22)
3. Завадський І.О. Основи баз даних: [Навч. посіб.] / І.О. Завадський. — К.: Видавець І.О. Завадський, 2011. — 192 с.: іл.
4. [Електронний ресурс] <http://lib.mdpu.org.ua/e-book/vstup/L5.htm>
5. Як появиласьбазаданих[Електронний ресурс] Режим доступу: [www.URLhttps://dp.informator.ua/2016/12/13/kak-poyavilasbaza](http://www.dp.informator.ua/2016/12/13/kak-poyavilasbaza)(дата звернення:15.06.22)
6. Моделі даних. Ієрархічна модель даних. Мережна модель даних. Реляційна модель даних[Електроннийресурс]Режимдоступу:[www.URLhttp://geoknigi.com/book_view.php?id=589](http://www.geoknigi.com/book_view.php?id=589)(дата звернення:15.06.22)
7. Види та моделі баз даних [Електронний ресурс] Режим доступу: [www.URL http://poznauka.org/s41543t1.html](http://www.poznauka.org/s41543t1.html) (дата звернення:15.06.22)
8. Об'єктно - орієнтоване програмування (ООП)[Електронний ресурс] Режимдоступу:[www.URLhttp://m.programming.in.ua/programming/basisprogramming/25-database](http://www.m.programming.in.ua/programming/basisprogramming/25-database) (дата звернення:15.06.22)
9. Полтавський національний технічний університет ім. Ю. Кондратюка [Електронний ресурс] Режим доступу: [www.URL http://www.studfiles.ru/preview/5725740/page:3/](http://www.studfiles.ru/preview/5725740/page:3/) (дата звернення:15.06.22)
10. Переменные и типы данных в С++ — урок 2 [Електронний ресурс] Режим доступу: [www.URL https://code-live.ru/post/cpp-variables-and-datatypes/](https://code-live.ru/post/cpp-variables-and-datatypes/) (дата звернення:15.06.22)
11. Основи програмування на С# для початківців [Електронний ресурс] Режим доступу: [www.URL http://purecodcpp.com/uk/](http://purecodcpp.com/uk/)(дата звернення:15.06.22)

12. Реляційна модель даних [Електронний ресурс] Режим доступу: www.URL:https://uk.wikipedia.org/wiki/Реляційна_модель_даних(дата звернення:15.06.22)

13. Створення бази даних "Відвідування" засобами Access [Електронний ресурс] Режим доступу: <https://studizba.com/files/show/doc/54467-1-49155.html>(дата звернення:14.06.22)

14. ІНФОРМАЦІЙНІ СИСТЕМИ ТА РЕЛЯЦІЙНІ БАЗИ ДАНИХ Навчальний посібник для студентів математичних спеціальностей [Електронний ресурс]Режимдоступу:https://dspace.uzhnu.edu.ua/jspui/bitstream/lib/19776/1/Мулеса_БД.pdf(дата звернення:15.06.22)

15. Теорія та практика прикладного програмування спеціальностей [Електронний ресурс] Режим доступу: <https://infourok.ru/kurovaya-rabota-na-temu-baza-danih-teoriya-ta-praktika-prikladnogo-programuvannya-4159270.html>(дата звернення:14.06.22)

16. Гожий О.П., Калініна І.О. Інформатика та комп'ютерна техніка: Навчально-методичний посібник. Для самостійного вивчення. – Миколаїв: Вид-во МДГУ ім. Петра Могили, 2006.Вип. 58. – 212 с.

