**КОЛОМИЙСЬКИЙ ПОЛІТЕХНІЧНИЙ КОЛЕДЖ**

**НАЦІОНАЛЬНОГО УНІВЕРСИТЕТУ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»**

**Циклова комісія інженерія програмного забезпечення**

**КУРСОВА РОБОТА**

з дисципліни: *Бази даних*

на тему: *Автоматизоване робоче місце управляючого авіакомпанії*

Студента 3 курсу групи П-31

галузі знань 12

спеціальності 121

Стоцький А.-В. О.

(прізвище та ініціали)

Керівник: Красничук В.М.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Оцінка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Члени комісії: \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(підпис) (прізвище та ініціали)

м. Коломия – 2018 рік

**Зміст**

Змн.

Арк.

№ докум.

Підпис

Дата

Арк.

1

419 КР 121.01.18 ПЗ

Розроб.

Стоцький А.-В. О.

Перевір.

Красничук В.М.

Реценз.

Н. Контр.

Затверд.

Пояснювальна записка.

Автоматизоване робоче місце управля-ючого авіакомпанії

Літ.

Аркушів

*26*

КПК НУ «ЛП»

Перелік скорочень………………………………………………………………...0

Вступ……………………………………………………………………………….1

1.Аналіз предметної області,короткий опис……………………………..……...2

1.1.Визначення необхідності створення ПЗ………………………………2

1.2.Постановка задачі………………………………………………………2

1.3.Короткий аналіз ПО……………………………………………………2

2.Семантичне моделювання предметної області……………………..…………3

2.1.Визначення сутностей,атрибутів та зв’язків між ними……….…….3

2.2.Складання діаграми сутностей……………………………….………3

2.3.Модель представлення даних "сутність-зв'язок"

(Entity-Relation модель)………………………………………………...…6

3.Розробка додатку для роботи з ПЗ……………………………………………8

3.1.Структурна схема додатку……………………………………………8

3.2.Розробка інформаційної структури даних, створення таблиць БД………………………………………………………………..…………………9

3.3.Побудова схеми даних ПО………………………….………………..10

3.4.Створення SQL запитів до таблиць БД……………….…………….11

3.5.Розробка необхідних звітів даної ПО………………….…………...14

3.6.Розробка форм користувача…………………………………………16

3.7.Створення програмного модуля та розробка програмного забезпечення необхідного для пошуку даних……………………………..20

3.8.Алгоритм та блок-схема пошуку даних……………………………22

Висновки………………………………………………………………………....25

Література………………………………………………………………………..26

**Перелік скорочень**

БД-база даних

СУБД – система управління базою даних

ПЗ-програмне забезпечення

ПО-предметна область

ПП- програмний продукт

ER-Entity Relation

**Вступ**

Метою курсової роботи є розробка програмного забезпечення для роботи авіакомпанії.

Авіакомпанії - підприємства, що пропонують послуги пов'язані з перевезеннями пасажирів або вантажів. На даний час цей повітряний транспорт все більше набуває популярності.

Найважливішою перевагою подорожей літаком є його висока швидкість пересування. Всього за кілька годин можна перетнути тисячі кілометрів і опинитися в потрібному вам місці. Особливо це зручно при обмеженому часі. Подорож літаком не здається такою нудною, як, наприклад, на поїзді або автобусі. Адже тут не потрібно декілька днів витрачати на поїздку. Літаком можна відправитися в будь-яку країну, куди не завжди можна дістатися залізничним або автомобільним транспортом.  Дуже вражає вид з ілюмінатора, красиві пейзажі, розташовані внизу або білосніжні хмари, що огортають літак. І саме авіакомпанії можуть надати можливість людям подорожувати повітряним транспортом.

**1. Аналіз предметної області, короткий опис**

**1.1. Визначення необхідності створення ПЗ**

На даний час подорож повітряним транспортом все більше набуває популярності, створюється все більше авіакомпаній. Через те що існує велика кількість рейсів, пасажирів, потрібно запам’ятовувати велику кількість інформації, тому необхідно створення надійної бази даних яка значно полегшить роботу управляючого авіакомпанії.

**1.2. Постановка задачі**

Створити ПЗ, яке дає змогу записувати, змінювати, видаляти дані щодо рейсів, яким літаком буде користуватися даний політ, які пасажири купили квиток на нього, який персонал обслуговує його.

**1.3. Короткий аналіз ПО**

Авіакомпанії пропонують людям зручний та вчасний переліт. Під час польоту пасажир відчуває себе максимально комфортно і зручно, що забезпечується зручними сидіннями, що дозволяють чудово відпочити і навіть поспати.

Плануючи свою відпустку, люди з особливою увагою вибирають найбільш відповідний транспорт, який доставить їх до місця відпочинку. Способів існує не так вже й багато — автомобільний, залізничний, водний і повітряний траспорт. Останній вид є найбільш молодим, але вже досить затребуваним і зарекомендувавшим себе.

Кожній авіакомпанії належить певна кількість літаків, своїх чи орендованих, які можуть вміщати в себе велику кількість пасажирів та багажу, кожен літак має різні характеристики.

Персонал літака переважно складається з командиру екіпажу, пілота, другого пілота, бортінженера, штурман, та трьох стюардеси. Перед польотом кожен пасажир повинен зареєструватися, передавши свої дані працівнику аеропорту, який внесе пасажира в базу даних.

**2. Семантичне моделювання предметної області.**

**2.1. Визначення сутностей, атрибутів та зв’язків між ними**

Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку [предметну область](https://uk.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D0%B4%D0%BC%D0%B5%D1%82%D0%BD%D0%B0_%D0%BE%D0%B1%D0%BB%D0%B0%D1%81%D1%82%D1%8C). Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними.

Сутність — це збірне поняття, деяка абстракція реально існуючого об'єкта, процесу, явища чи деякого уявлення про об'єкт.

Атрибут – елемент даних, що описує певну властивість сутності. Кожна сутність має мати мінімальний набір унікальних атрибутів, що зветься первинним ключем.

Зв’язок — це асоціація між двома або більше сутностями, що є суттєвою для аналізованої предметної області, забезпечує взаємодію між сутностями.

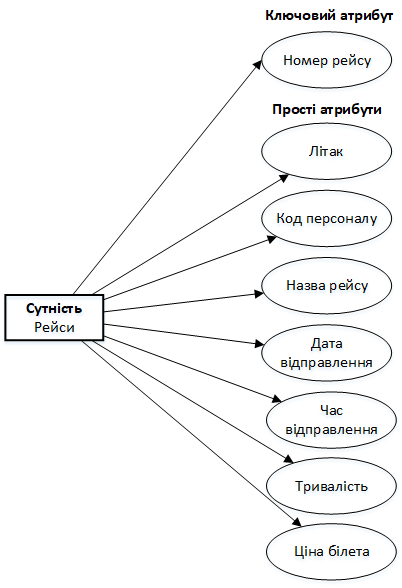
**2.2. Складання діаграм сутностей**

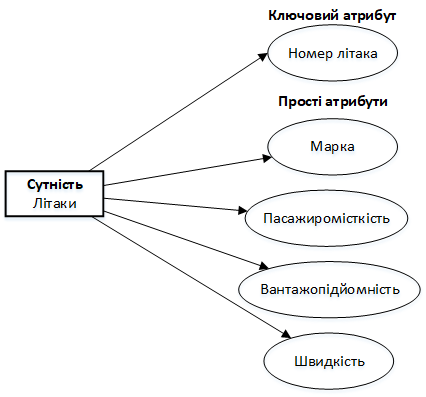
|  |
| --- |
| **Сутність:**  Рейси |
|  |
| **Атрибути:** |
| Номер рейсу |
| Літак |
| Код персоналу |
| Назва рейсу |
| Дата відправлення |
| Час відправлення |
| Тривалість |
| Ціна білета |

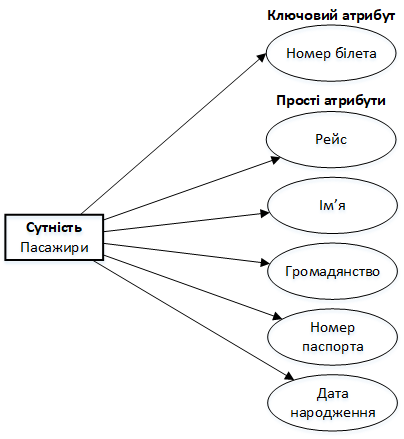
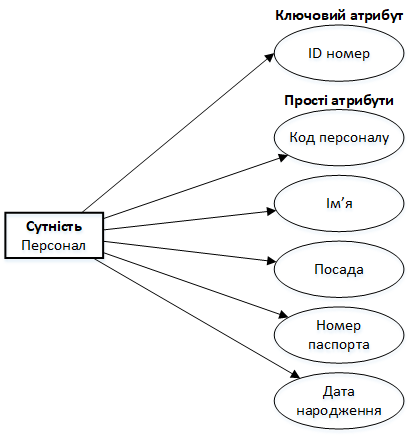
|  |
| --- |
| **Сутність:**  Персонал |
|  |
| **Атрибути:** |
| ID номер |
| Код персоналу |
| Ім’я |
| Посада |
| Номер паспорта |
| Дата народження |

|  |
| --- |
| **Сутність:**  Літак |
|  |
| **Атрибути:** |
| Номер літака |
| Марка |
| Пасажиромісткість |
| Вантажопідйомність |
| Швидкість |

|  |
| --- |
| **Сутність:**  Пасажири |
|  |
| **Атрибути:** |
| Номер білета |
| Рейс |
| Ім’я |
| Громадянство |
| Номер паспорта |
| Дата народження |

****

****

**2.3. Модель представлення даних «сутність-зв’язок» (Entity‑Relation модель)**

Модель сутність-зв'язок (ER-модель) - модель даних, що дозволяє описувати концептуальні схеми предметної області. ER-модель використовується при концептуальному проектуванні баз даних. З її допомогою можна виділити ключові сутності і позначити зв'язки, які можуть встановлюватися між цими сутностями.

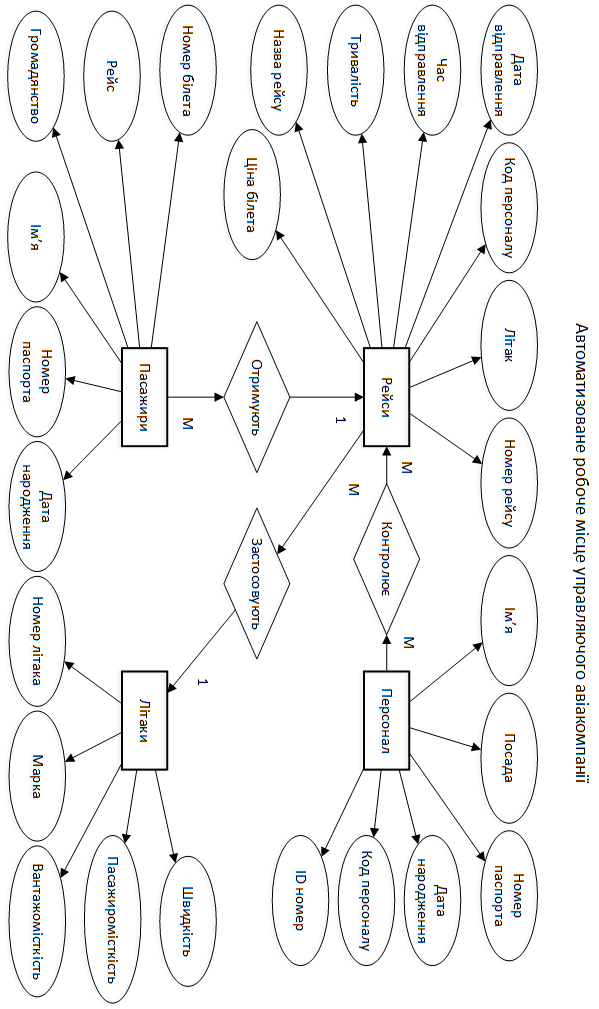
Під час проектування баз даних відбувається перетворення ER‑моделі в конкретну схему бази даних на основі обраної моделі даних. ER‑модель являє собою формальну конструкцію, яка сама по собі не наказує ніяких графічних засобів її візуалізації. Як стандартна графічна нотація, за допомогою якої можна візуалізувати ER-модель, була запропонована діаграма сутність-зв'язок (ER‑діаграма).

Сутності та зв'язки, представлені в ER-моделі, повинні бути перетворені в інформаційні компоненти бази даних - таблиці, стовпці, обмеження цілісності, які утворюють так звану серверну модель.

Умовам цілісності відповідають:

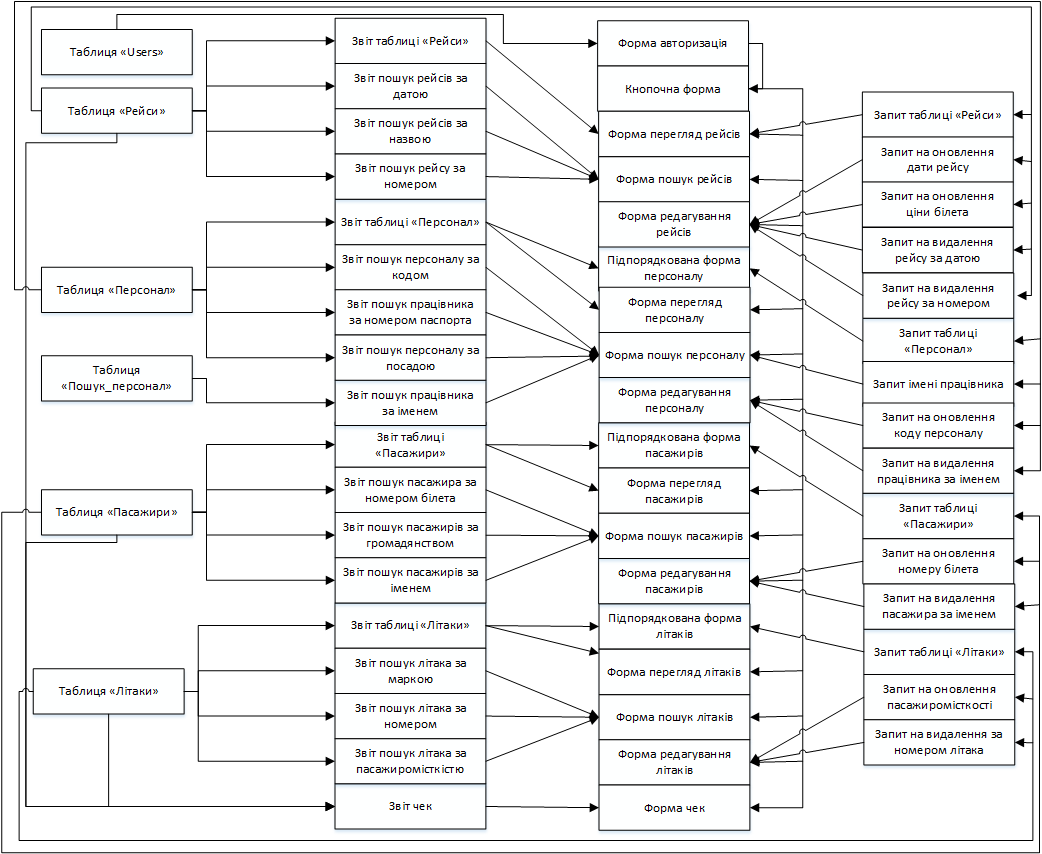
- Ключі трьох типів (первинний, унікальний і зовнішній),

- А також властивість обов'язковості (визначеності) значень атрибутів-стовпців, що утворюють ключі.

****

**3. Розробка додатку для роботи з ПЗ**

**3.1. Структурна схема додатку**

****

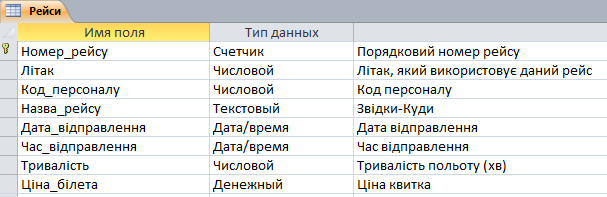
**3.2. Розробка інформаційної структури даних, створення таблиць БД**

**Структура таблиць з описом полів.**

**Таблиця «Рейси»**

SQL запит:

CREATE TABLE Reisy(Nomer\_reisu INT PRIMARY KEY, Litak INT, Kod\_personalu INT, Nazva\_reisu VARCHAR(30), Data\_vidpravlennia DATE, Chas\_vidpravlennia TIME, Trivalis INT, Cina\_bileta INT);

****

**Таблиця «Літаки»**

SQL запит:

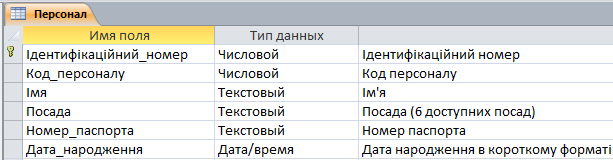
CREATE TABLE Litaku(Nomer\_litaka INT PRIMARY KEY, Marka VARCHAR(30), Pasagiromistkist INT, Vantagopidyomnist INT, Shvidkist INT);

****

**Таблиця «Персонал»**

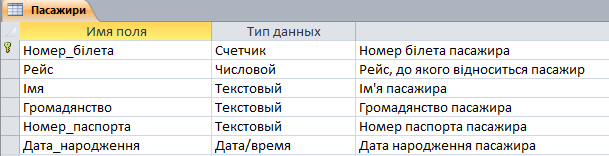
SQL запит:

CREATE TABLE Personal(ID INT PRIMARY KEY, Kod\_personalu INT, Imya VARCHAR(50), Posada VARCHAR(20), Nomer\_pasporta VARCHAR(30), Data\_narodzennia DATE);

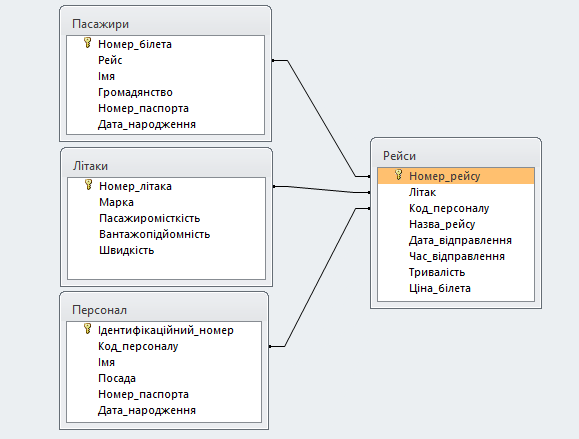
****

**Таблиця «Пасажири»**

SQL запит: CREATE TABLE Passagiri(Nomer\_bileta INT PRIMARY KEY, Reis INT, Imya VARCHAR(50), Gromadianstvo VARCHAR(20), Nomer\_pasporta VARCHAR(10), Data\_narodzennia DATE);

****

**3.3. Побудова схеми даних ПО**

****

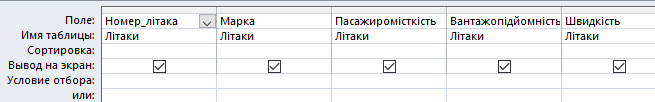
**3.4. Створення SQL запитів до таблиць БД**

1. Запит на вибірку. Переглядаємо записи таблиці «Літаки».

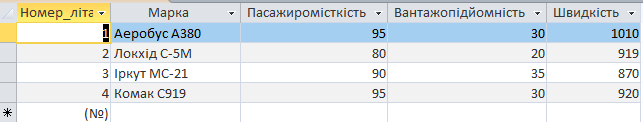
SQL код:

SELECT Літаки.Номер\_літака, Літаки.Марка, Літаки.Пасажиромісткість, Літаки.Вантажопідйомність, Літаки.Швидкість

FROM Літаки;

Створення запиту:  


Результат роботи:



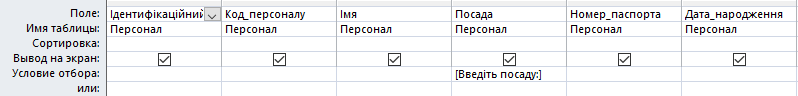
2. Запит з параметром. Вводимо посаду, отримуємо інформацію про людей з такою посадою.

SQL код:  
SELECT Персонал.Ідентифікаційний\_номер, Персонал.Код\_персоналу, Персонал.Імя, Персонал.Посада, Персонал.Номер\_паспорта, Персонал.Дата\_народження

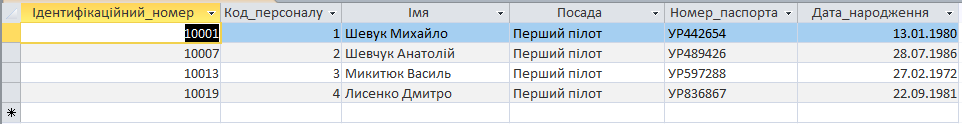
FROM Персонал

WHERE (((Персонал.Посада)=[Введіть посаду:]));

Створення запиту:



Результат роботи:



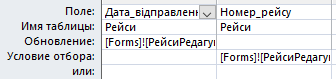
3. Запит на оновлення. Вводимо для вибраного рейсу нову дату відправлення.

SQL код:

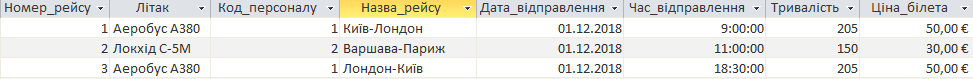
UPDATE Рейси SET Рейси.Дата\_відправлення = [Forms]![РейсиРедагування]![data]

WHERE (((Рейси.Номер\_рейсу)=[Forms]![РейсиРедагування]![nomer]));

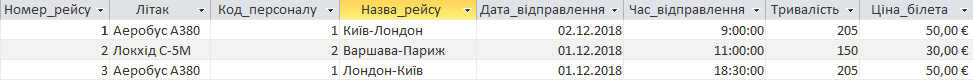
Створення запиту:



Результат роботи:

До:

Після:



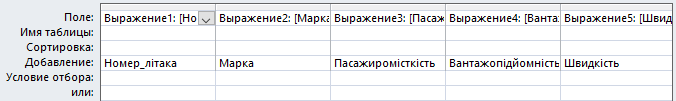
4. Запит на добавлення. Добавляємо новий запис в таблицю Літаки.

SQL код:

INSERT INTO Літаки ( Номер\_літака, Марка, Пасажиромісткість, Вантажопідйомність, Швидкість )

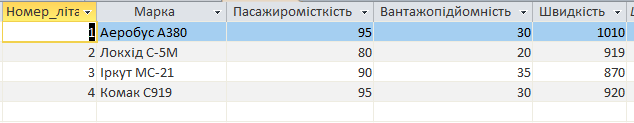
SELECT [Номер літака:] AS Выражение1, [Марка літака:] AS Выражение2, [Пасажиромісткість:] AS Выражение3, [Вантажопідйомність:] AS Выражение4, [Швидкість:] AS Выражение5

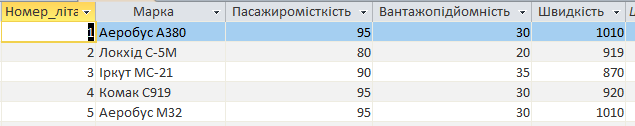
FROM Літаки;

Створення запиту:

Результат роботи:

До:



Після:

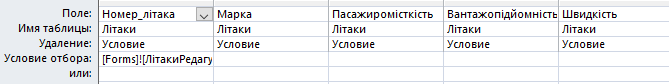
5. Запит на видалення. Видаляємо запис про літак за номером.

SQL код:

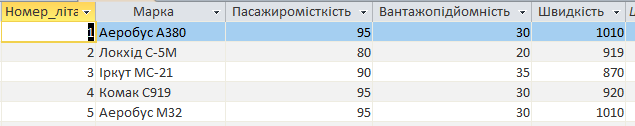
DELETE Літаки.Номер\_літака, Літаки.Марка, Літаки.Пасажиромісткість, Літаки.Вантажопідйомність, Літаки.Швидкість

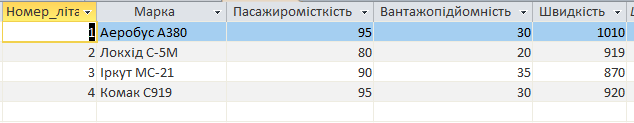
FROM Літаки

WHERE (((Літаки.Номер\_літака)=[Forms]![ЛітакиРедагування]![nomer2]));

Створення запиту:

Результат роботи:

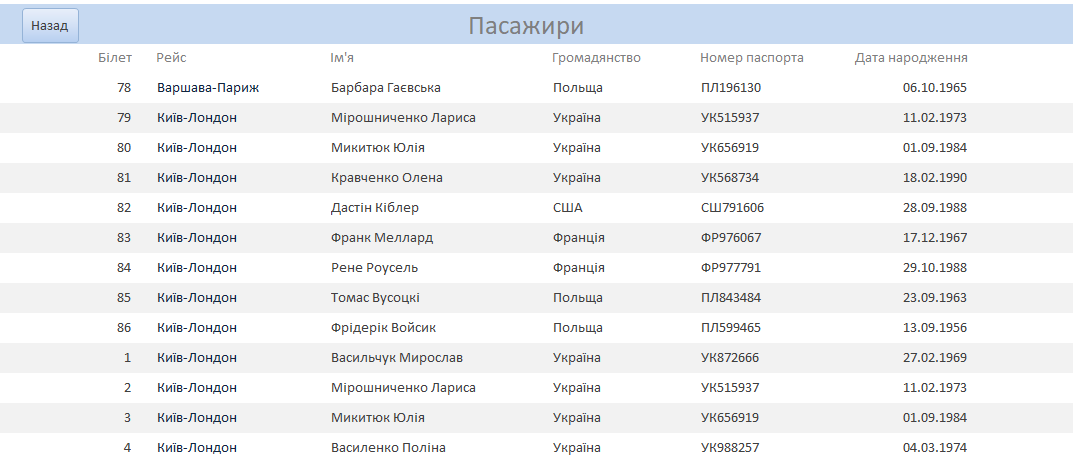
До: 

Після: 

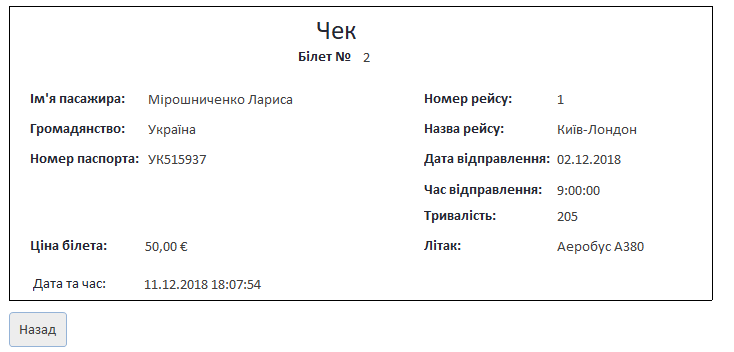
**3.5. Розробка необхідних звітів даної ПО**

1. Звіт до таблиці «Рейси»

2. Звіт до таблиці «Персонал»

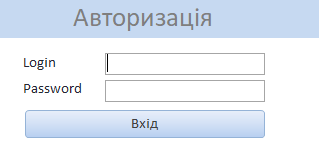
3. Звіт до таблиці «Пасажири»

4. Звіт до таблиці «Літаки»

5. Звіт чеку

**3.6. Розробка форм користувача**

1. Форма авторизації

****

Програмний модуль до форми:

Private Sub p\_AfterUpdate()

Call auth(Forms!Authorization!l, Forms!Authorization!p)

End Sub

Private Sub Кнопка7\_Click()

Call auth(Forms!Authorization!l, Forms!Authorization!p)

End Sub

Програмний модуль авторизації:

Function auth(login, password)

Dim db As DAO.Database

Dim pk As Recordset

Dim a As String

Set db = CurrentDb()

Set pk = db.OpenRecordset("Select \* FROM users")

Do Until pk.EOF

If login = pk!user\_name And password = pk!user\_password Then

DoCmd.Close

DoCmd.OpenForm "Головна\_форма"

a = "ok"

Exit Do

Else: a = "no"

End If

pk.MoveNext

Loop

If a = "no" Then

MsgBox ("Хибний логін або пароль")

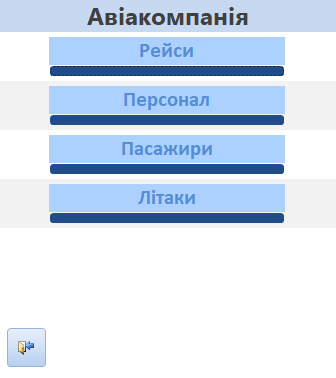
End If

pk.Close

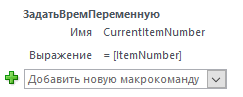
Set db = Nothing

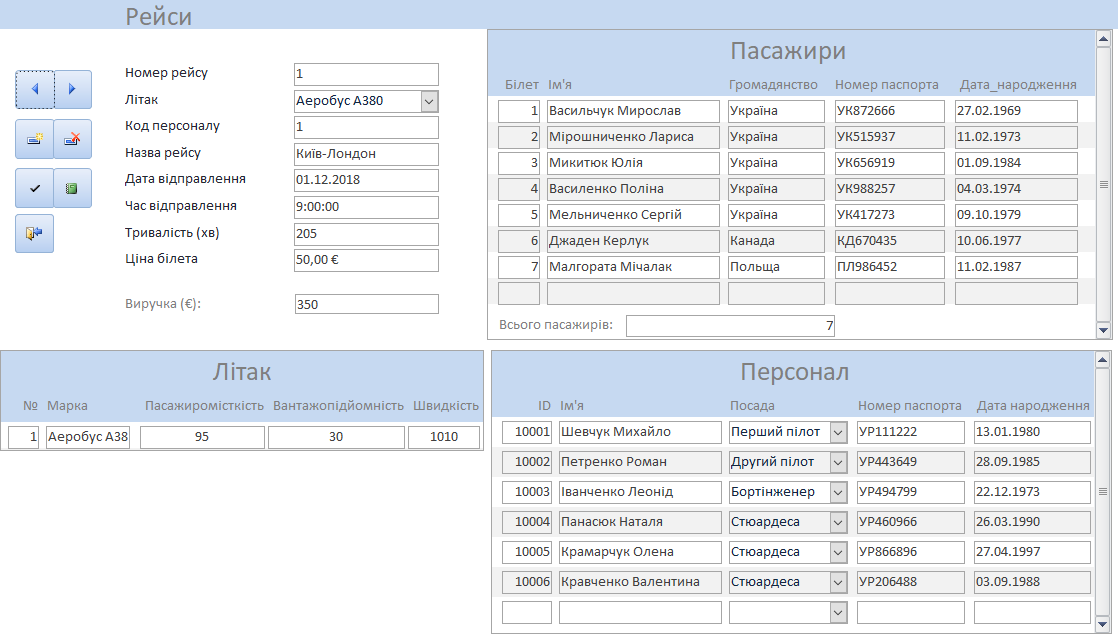
End Function

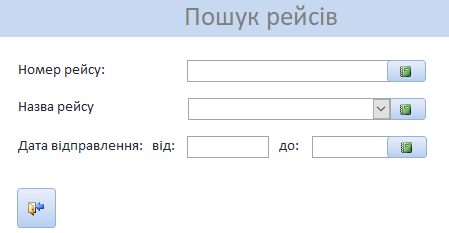
2. Головна кнопочна форма

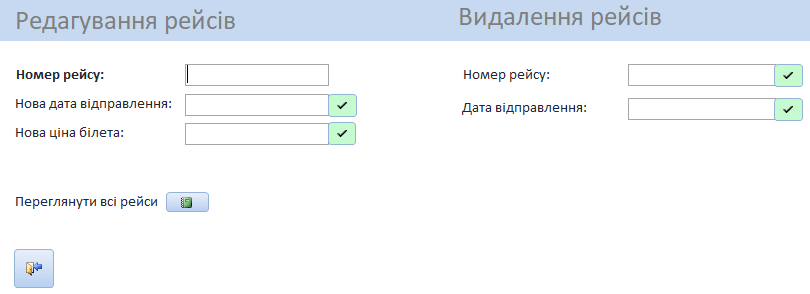
****

Макрос до форми:

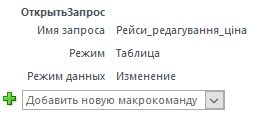
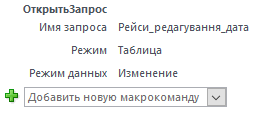
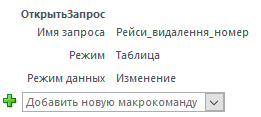
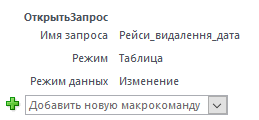


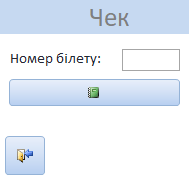
3. Форма перегляду рейсів ****

4. Форма пошуку рейсів ****

5. Форма редагування рейсів ****

Макроси до форми:

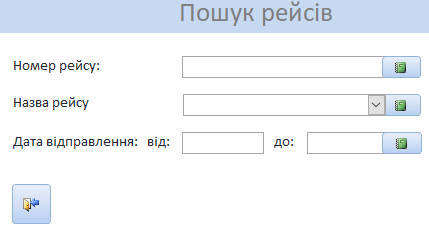


6. Форма перегляду чеку ****

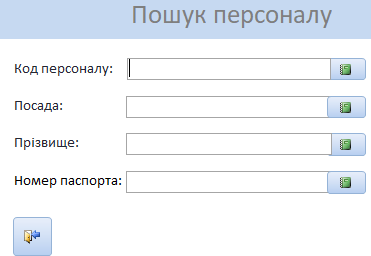
**3.7. Створення програмного модуля та розробка програмного забезпечення необхідного для пошуку даних**

1. Форми для пошуку даних

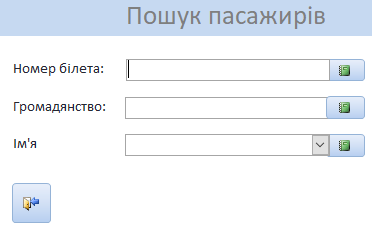
Форма пошуку рейсів



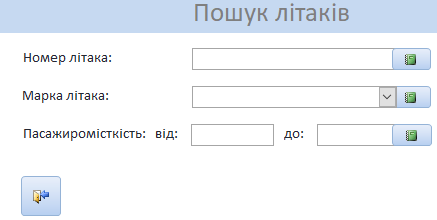
Форма пошуку персоналу



Форма пошуку пасажирів

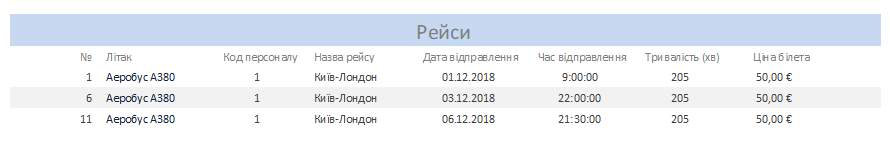


Форма пошуку літаків



2. Звіти результатів пошуку

Звіт результатів пошуку рейсів



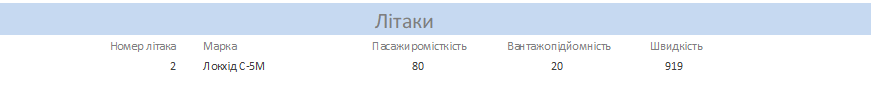
Звіт результатів пошуку персоналу



Звіт результатів пошуку пасажирів



Звіт результатів пошуку літаків



**3.8. Алгоритм та блок-схема пошуку даних**

Алгоритм пошуку даних:

Function search(par) **'створюємо функцію, яка буде приймати параметр par**

Dim db As DAO.Database **'описуємо об'єктну змінну класу Database**

Dim pk As Recordset **'описуємо об'єктну змінну класу Recordset**

Dim pk1 As Recordset **'описуємо об'єктну змінну класу Recordset**

Dim pk2 As Recordset **'описуємо об'єктну змінну класу Recordset**

Set db = CurrentDb() **'присвоюємо об'єкту db класу Database значення текучої бази даних**

Set pk = db.OpenRecordset("SELECT \* FROM Пошук\_персонал") **'присвоєння об'єктній змінній таблиці Пошук\_персонал**

Set pk1 = db.OpenRecordset("SELECT \* FROM Персонал") **'присвоєння об'єктній змінній таблиці Персонал**

Set pk2 = db.OpenRecordset("SELECT \* FROM Пасажири") **'присвоєння об'єктній змінній таблиці Пасажири**

db.Execute "DELETE \* FROM Пошук\_персонал" **'видаляємо всі записи з таблиці Пошук\_персонал**

Do Until pk1.EOF **'запускаємо цикл, поки не закінчуться записи в p1**

If par = pk1!Номер\_паспорта Then **'прирівнюємо параметр до поля Номер\_паспорта об'єкта pk1**

pk.AddNew **'створення нового запису для об'єкта pk класу Recordset**

pk!Ідентифікаційний\_номер = pk1!Ідентифікаційний\_номер **'присвоюємо полю Ідентифікаційний\_номер об'єкта pk значення поля Ідентифікаційний\_номер об'єкта pk**

pk!Код\_персоналу = pk1!Код\_персоналу **'присвоюємо полю Код\_персоналу об'єкта pk значення поля Код\_персоналу об'єкта pk1**

pk!Імя = pk1!Імя **'присвоюємо полю Імя об'єкта pk значення поля Імя об'єкта pk1**

pk!Посада = pk1**!Посада 'присвоюємо полю Посада об'єкта pk значення поля Посада об'єкта pk1**

pk!Номер\_паспорта = pk1!Номер\_паспорта **'присвоюємо полю Номер\_паспорта об'єкта pk значення поля Номер\_паспорта об'єкта pk1**

pk!Дата\_народження = pk1!Дата\_народження **'присвоюємо полю Дата\_народження об'єкта pk значення поля Дата\_народження об'єкта pk1**

pk.Update **'оновлення полів таблиці**

End If **'кінець оператора if**

pk1.MoveNext **'метод об'єкту pk1 класу Recordset (перехід до наступного запису)**

Loop **'кінець циклу**

pk1.Close **'закриття таблиці pk1**

Do Until pk2.EOF **'запускаємо цикл, поки не закінчуться записи в p1**

If par = pk2!Номер\_паспорта Then **'прирівнюємо параметр до поля Номер\_паспорта об'єкта pk2**

pk.AddNew **'створення нового запису для об'єкта pk класу Recordset**

pk!Імя = pk2!Імя **'присвоюємо полю Імя об'єкта pk значення поля Імя об'єкта pk2**

pk!Номер\_паспорта = pk2!Номер\_паспорта **'присвоюємо полю Номер\_паспорта об'єкта pk значення поля Номер\_паспорта об'єкта pk2**

pk.Update **'оновлення полів таблиці**

End If **'кінець оператора if**

pk2.MoveNext **'метод об'єкту pk2 класу Recordset (перехід до наступного запису)**

Loop **'кінець циклу**

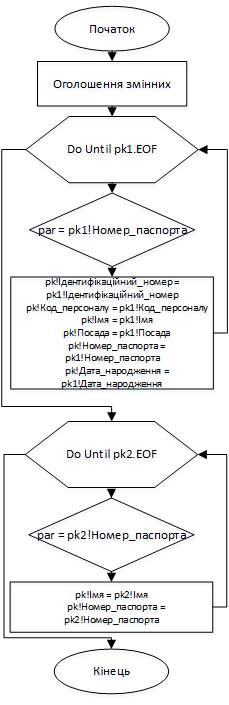
pk2.Close **'закриття таблиці pk2**

pk.Close **'закриття таблиці pk**

Set db = Nothing **'встановлення значення Nothing в об'єктну змінну db**

End Function **'кінець функції**

Блок-схема алгоритму:



**ВИСНОВОК**

Результатом курсової роботи стала реляційна модель бази даних, яка була створена на основі концептуальної моделі вибраної предметної області. В розробленому додатку ми можемо працювати з даними, переглядати, добавляти, видаляти, змінювати. За допомогою VBA був розроблений пошук даних. Зручний інтерфейс дає змогу без зайвих зусиль орієнтуватися в програмі.

Розроблений додаток відповідає всім вимогам предметної області, а так само каталогу завдань і запитів. Всі поставлені цілі були виконані повністю.

Дана база даних і додаток було розроблено з метою полегшити роботу з великою кількістю інформації для авіакомпанії та автоматизувати робоче місце управляючого. У зв'язку з тим, що база навчальна, а не професійна, деякі дані не були включені у базу. Але розроблена в даній роботі база даних легко доповнюється за необхідності розробки професійної бази даних.

**ЛІТЕРАТУРА**

1. Хомоненко А.Д. Бази даних: підручник для ВНЗ / А.Д. Хомоненко, В.М. Циганков, Авт. Мальцев. - 4-те вид., доп. і перероб. - СПб.: КОРОНАпринт, 2004. - 736 с.

2. Програмирование для Microsoft Access 2000 за 24 часа, «Вильямс» Москва, 2000

3. Саак А.Е. Інформаційні технології управління: підручник для вузів / А.Е.Саак, Є.В. Пахомов, В.М. Тюшняков. - СПб.: Пітер, 2005. - 320 с.

4. Основні відомості про бази даних

<https://support.office.com/uk-ua/article/Основні-відомості-про-бази-даних-a849ac16-07c7-4a31-9948-3c8c94a7c204#__toc257378513>

5. Разработка корпоративных приложений в Access 2002. Для профессионалов, Питер, BHV, Год:2003

6. Робота в системі управління базами даних MS ACCESS

https://stud.com.ua/62417/menedzhment/robota\_sistemi\_upravlinnya\_bazami\_danih\_access