

Міністерство освіти і науки України
Чернівецький національний університет
Імені Юрія Федьковича
Інститут фізико-технічних та комп'ютерних наук
Відділ комп'ютерних технологій
Кафедра математичних проблем управління і кібернетики

ЗВІТ
про виконання практичної роботи
з дисципліни «Організація баз даних і знань»

Виконав студент

Бужак А.В.

Курс II

Група 241

Викладач

Спіжавка Д.І.

Чернівці – 2018

Завдання

20. Інформаційна система аеропорту

Працівників аеропорту можна розділити на пілотів, диспетчерів, техніків, касирів, робітників служби безпеки, довідкової служби й інших, що адміністративно відносяться кожний до свого відділу. Кожна з перерахованих категорій працівників має унікальні атрибути-характеристики, обумовлені професійною спрямованістю. У відділах існує розбиття робітників на бригади. Відділи очолюються начальниками, які представляють собою адміністрацію аеропорту. У функції адміністрації входить планування рейсів, складання розкладів, формування кадрового складу аеропорту. За кожним літаком закріплюється бригада пілотів, техніків й обслуговуючого персоналу. Пілоти зобов'язані проходити щороку медогляд, тих, що не пройшли медогляду необхідно перевести на іншу роботу. Літак повинен своєчасно оглядатися техніками й при необхідності ремонтуватися. Підготовка до рейсу містить у собі технічну частину (техогляд, заправка необхідної кількості палива) і обслуговуючу частину (прибирання салону, запас продуктів харчування й т.п.).

В розкладі вказується тип літака, рейс, дні відльоту, час відльоту й прильоту, маршрут (початковий і кінцевий пункти призначення, пункт пересадки), вартість квитка. Квитки на авіарейси можна придбати завчасно або забронювати в авіакасах. Ціна квитка залежить не тільки від маршруту, але й від часу відльоту (у незручний час – вночі або рано вранці – ціна квитка нижче). До відправлення рейсу, якщо в цьому є необхідність, квиток можна повернути. Авіарейси можуть бути затримані через погодні умови, технічні неполадки, а також можуть бути скасовані, якщо продано менше встановленого мінімуму квитків.

Авіарейси можна розділити на наступні категорії: внутрішні, міжнародні, чартерні, вантажоперевезення, спеціальні рейси. Під час посадки в літак пасажир повинен пред'явити квиток та паспорт, а для міжнародного рейсу, окрім вказаного, має пройти митний контроль. Пасажири можуть здавати свої речі в багажне відділення. На рейси вантажоперевезень і спеціальні рейси квитки не продаються. Для спец. рейсів не існує розкладу. Квитки на чартерні рейси розповсюджує те агентство, яке його організувало.

Види запитів в інформаційній системі:

- 1) Одержати список і загальне число всіх робітників аеропорту, начальників відділів, робітників зазначеного відділу, за стажем роботи в аеропорту, статевую ознакою, віком, ознакою наявності й кількості дітей, за розміром заробітної платні.
- 2) Одержати перелік і загальне число робітників у бригаді, по всіх відділах, у зазначеному відділі, що обслуговують конкретний рейс, за віком, сумарною(середньою) зарплатнею у бригаді.
- 3) Одержати перелік і загальне число пілотів, що пройшли медогляд або не пройшли його в зазначений рік, за статевую ознакою, віком, розміром заробітної платні.
- 4) Одержати перелік і загальне число літаків приписаних до аеропорту, що перебувають у ньому в зазначений час, за часом надходження в аеропорт, за кількістю здійснених рейсів.
- 5) Одержати перелік і загальне число літаків, що пройшли техогляд за певний період часу, відправлених у ремонт у зазначений час, ремонтваних задане число разів, за кількістю здійснених рейсів до ремонту, за віком літака.
- 6) Одержати перелік і загальне число рейсів по зазначеному маршруту, за тривалістю перельоту, за ціною квитка й по всім цим критеріям разом.
- 7) Одержати перелік і загальне число скасованих рейсів повністю, у зазначеному напрямку, по зазначеному маршруту, за кількістю невикористаних місць, за відсотковим співвідношенням невикористаних місць.
- 8) Одержати перелік і загальне число затриманих рейсів повністю, за зазначеною причиною, по зазначеному маршруту, і кількість зданих квитків під час затримки.
- 9) Одержати перелік і загальне число рейсів, якими літають літаки заданого типу й середню кількість проданих квитків на визначені маршрути, за тривалістю перельоту, за ціною квитка, часом вильоту.
- 10) Одержати перелік і загальне число авіарейсів зазначеної категорії, у певному напрямку, із зазначеним типом літака.
- 11) Одержати перелік і загальне число пасажирів на даному рейсі, що полетіли в зазначений день, що полетіли за кордон в зазначений день, за ознакою здачі речей у багажне відділення, за статевую ознакою, за віком.

12) Одержати перелік і загальне число вільних і заброньованих місць на зазначеному рейсі, на конкретний день, по зазначеному маршруту, за ціною, за часом вильоту.

13) Одержати загальне число зданих квитків на деякий рейс, у зазначений день, по певному маршруту, за ціною квитка, за віком, статтю.

Хід роботи

Проектування бази даних. Аналізуючи завдання, можна виділити такі сутності предметної області і їх атрибути:

I. Адміністрація

- ВідділиID
- ПрацівникиID

II. Бригади

- ID
- Назва

III. Відділи

- ID
- Назва

IV. ЖурналМедоглядів

- ID
- ПрацівникиID
- Дата
- Результат

V. ЖурналТехоглядів

- ID
- ЛітакиID
- ДатаОгляду
- РезультатТехогляду

VI. Квитки

- ID
- РейсиID
- ПасажириID

- СтатусКвиткаID
- НаявністьБагажу

VII. Літаки

- ID
- Номер
- ТипиЛітаківID
- ДатаВипуску
- КількістьМісць
- БригадиID

VIII. Маршрути

- ID
- ПочатковийПункт
- ПунктПересадки
- КінцевийПункт

IX. Пасажири

- ID
- ПІБ
- Паспорт
- Стать
- ДатаНародження

X. Працівники

- ID
- Прізвище
- Ім'я
- ПоБатькові

- ДатаНародження
- Стать
- КількістьДітей
- ДатаПрийомуНаРоботу
- ЗаробітнаПлата
- ВідділиID
- БригадиID

XI. Рейси

- ID
- ЛітакиID
- СтатусРейсуID
- ТипиРейсівID
- МаршрутиID
- ДатаЧасВідльоту
- ДатаЧасПрильоту
- ЦінаКвитка

XII. СтатусКвитка

- ID
- Статус

XIII. СтатусРейсу

- ID
- Статус

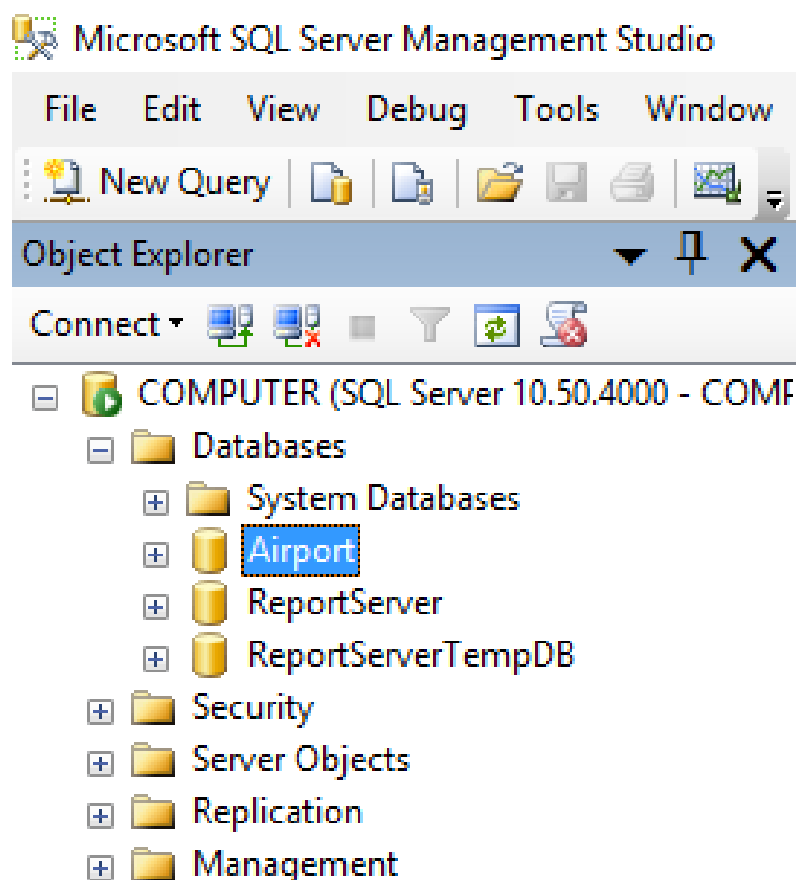
XIV. ТипиЛітаків

- ID
- Тип

XV. ТипиРейсів

- ID
- Тип

В середовищі **Microsoft SQL Server 2008 R2 Management Studio** створюю нову базу даних 'Airport'.



Надалі з базою даних можна працювати у середовищі **Microsoft SQL Server Management Studio** або **Microsoft Visual Studio**.

Уся інформація у базі даних зберігається в таблицях, які є засобом зберігання даних.

Таблиці складаються з рядків або записів і стовпців або полів. Кожне поле має три характеристики:

1. Ім'я поля - використовується для звернення до поля;
2. Значення поля - визначає інформацію, що зберігається в полі;
3. Тип даних поля - визначає, який вид інформації можна зберігати в полі.

У SQL сервер використовується наступні типи даних :

- Двійкові типи даних, які містять послідовності нулів і одиниць : 1) **binary**(n) — двійковий тип фіксованої довжини розміром в n байт, де n — значення від 1 до 8000; розмір при зберіганні складає n байт;

2) **varbinary**(n) - двійковий тип зі змінною довжиною, n може мати значення від 1 до 8000, max вказує, що максимальний розмір при зберіганні складає 23-1;

- Цілочисельні типи даних - типи даних для зберігання цілих чисел (у дужках вказаний діапазон значень типу даних) : **tinyint** (0..255), **smallint** (-32768..32767), **int** (-2147483648..2147483647), **bigint** (-2⁶³..2⁶³-1);

- Типи даних для зберігання чисел з плаваючою комою: **real** займає в пам'яті 4 байти; **float**(n), де n — ця кількість бітів, використовуваних для зберігання мантиси числа у форматі **float** при експоненціальному представленні, визначає точність даних і розмір для зберігання; значення параметра n повинне лежати в межах від 1 до 53; значенням за умовчанням для параметра n є 53;

- Типи даних для зберігання чисел з фіксованою точністю і масштабом **decimal**(p, s) і **numeric**(p, s), де p (точність) - максимальна кількість десяткових розрядів числа (як ліворуч, так і праворуч від десяткової коми). Точність має бути значенням в діапазоні від 1 до 38. За умовчанням це значення дорівнює 18. s (масштаб) - максимальна кількість десяткових розрядів числа праворуч від десяткової коми. Масштаб може набувати значення від 0 до p і може бути вказаний тільки спільно з точністю. За умовчанням масштаб набуває значення 0; тому $0 \leq s \leq p$;

- Символьні типи даних : **char**(n) - строковые дані фіксованої довжини не в Юнікоде, аргумент n визначає довжину рядка і повинен мати значення від 1 до 8000, розмір при зберіганні складає n байт; **varchar**(n | max) - строкові дані змінної довжини не в Юнікоде, аргумент n визначає довжину рядка і повинен

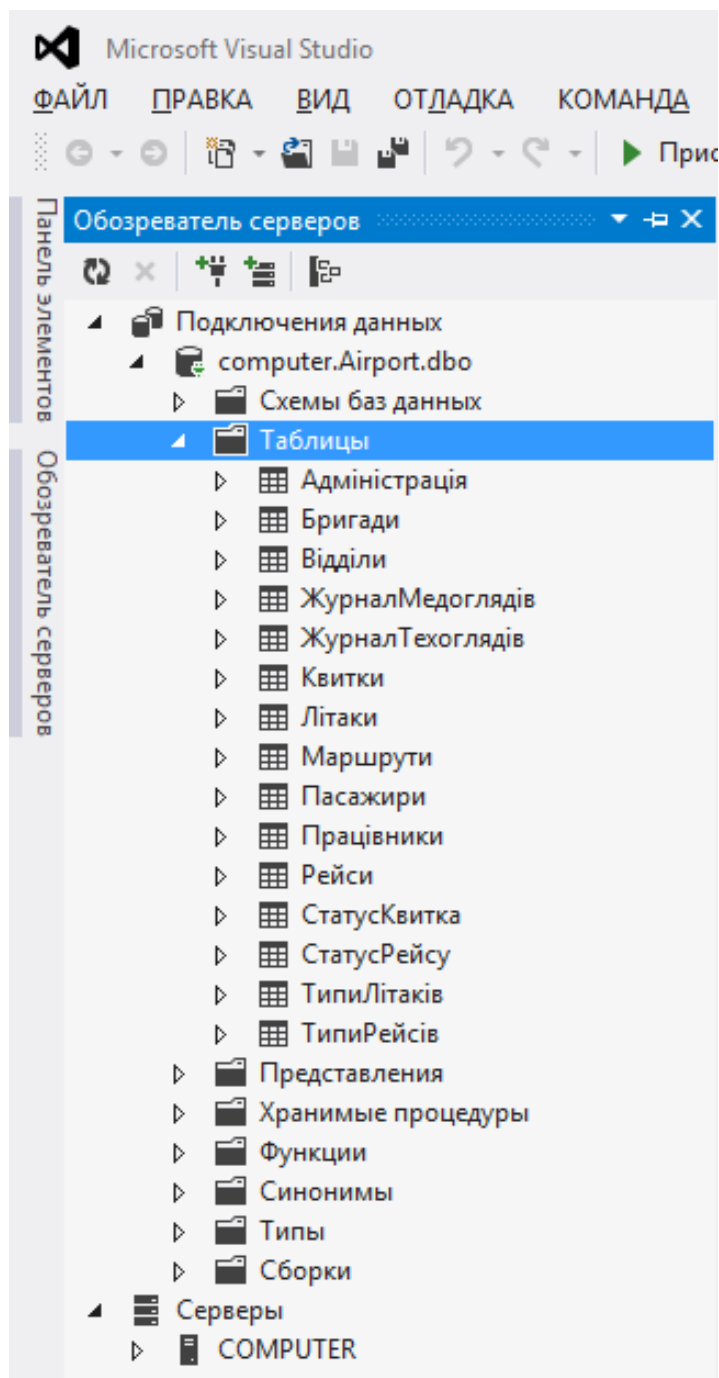
мати значення від 1 до 8000, значення max вказує, що максимальний розмір при зберіганні складає 231-1 байт (2 ГБ); text - дані змінної довжини не в Юнікоде в кодовій сторінці сервера і з максимальною довжиною рядка 231-1;

- Спеціальні типи даних : **bit** - цілочисельний тип даних, який може набувати значень 1, 0 або NULL; **image** - тип даних для зберігання малюнка розміром до 2ГБ;

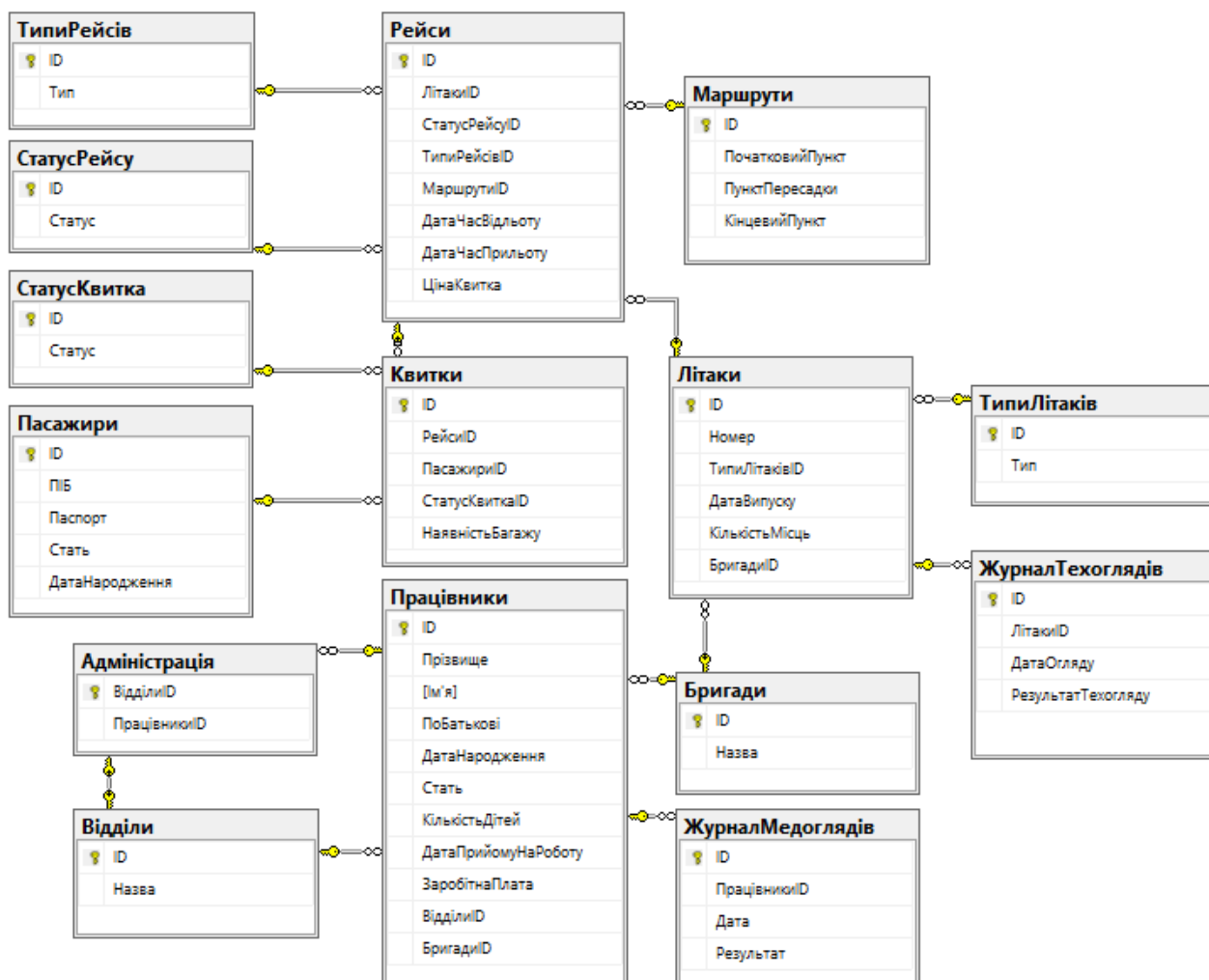
- Типи даних дати і часу : **date** (від 01.01.0001 до 31.12.9999); **datetime** (діапазон дати - від 01.01.1753 до 31.12.1999, діапазон часу - від 00:00:00 до 23:59:59, 997); **smalldatetime** (діапазон дати - від 01.01.1900 до 6.06.2079, діапазон часу - від 00:00:00 до 23:59:59); time (від 00:00:00.0000000 до 23:59:59.9999999);

- Грошові типи даних для зберігання фінансової інформації : **money** (8 байт) і **smallmoney** (4 байти) - типи даних, що представляють грошові (валютні) значення.

В базі даних створюю таблиці з полями, відповідно до початкового аналізу.



Реалізую діаграму зв'язків бази даних.



Реалізую задані запити в «Збережуваних процедурах» з відповідними вхідними параметрами, необхідними для виконання запиту.

Нижче наведено скрипти створення збережуваних процедур, які реалізують задані запити:

1. `ALTER PROCEDURE` dbo.[Query1]

```
(  
    @начальники nvarchar(3) = NULL,  
    @відділ nvarchar(25) = NULL,  
    @стать bit = NULL,  
    @кількістьДітейНижняМежа tinyint = 0,  
    @кількістьДітейВерхняМежа tinyint = 255,  
    @ЗПНижняМежа int = 0,  
    @ЗПВерхняМежа int = 1000000000,  
    @вікНижняМежа int = 0,  
    @вікВерхняМежа int = 200,  
    @стажНижняМежа int = 0,  
    @стажВерхняМежа int = 200  
)
```

`AS`

`SELECT`

```
[Прізвище], [Ім'я], [ПоБатькові], Відділи.Назва AS Відділ,  
[КількістьДітей], [ЗаробітнаПлата],  
DATEDIFF(month, [ДатаНародження], GETDATE()) / 12 AS Вік,  
DATEDIFF(month, [ДатаПрийомуНаРоботу], GETDATE()) / 12 AS
```

Стаж

```
FROM Працівники  
JOIN Відділи  
ON Працівники.ВідділиID = Відділи.ID  
WHERE [КількістьДітей] BETWEEN @кількістьДітейНижняМежа AND  
@кількістьДітейВерхняМежа  
AND [ЗаробітнаПлата] BETWEEN @ЗПНижняМежа AND @ЗПВерхняМежа  
AND DATEDIFF(month, [ДатаНародження], GETDATE()) / 12 BETWEEN  
@вікНижняМежа AND @вікВерхняМежа  
AND DATEDIFF(month, [ДатаПрийомуНаРоботу], GETDATE()) / 12  
BETWEEN @стажНижняМежа AND @стажВерхняМежа  
AND (CASE WHEN @начальники IS NOT NULL THEN  
          CASE WHEN (Працівники.ID IN (SELECT  
Адміністрація.ПрацівникиID FROM Адміністрація)) THEN  
          1  
          ELSE
```

```

                                0
                                END
                                ELSE
                                1
                                END) = 1
                                AND (CASE WHEN @Відділ IS NOT NULL THEN
                                CASE WHEN (Працівники.ВідділиID = (SELECT ID FROM
Відділи WHERE Назва = @Відділ)) THEN
                                1
                                ELSE
                                0
                                END
                                ELSE
                                1
                                END) = 1
                                AND (CASE WHEN @стать IS NOT NULL THEN
                                CASE WHEN (Працівники.Стать = @стать) THEN
                                1
                                ELSE
                                0
                                END
                                ELSE
                                1
                                END) = 1
                                ORDER BY Відділ, Прізвище, [Ім'я], Побатькові

RETURN

```

2. ALTER PROCEDURE dbo.Query2

```
(
    @бригада nvarchar(50) = NULL,
    @відділ nvarchar(25) = NULL,
    @рейс bigint = NULL,
    @вікНижняМежа tinyint = 0,
    @вікВерхняМежа tinyint = 200,
    @ЗПНижняМежа int = 0,
    @ЗПВерхняМежа int = 1000000000,
    @середняЗПуБригадіНижняМежа int = 0 ,
    @середняЗПуБригадіВерхняМежа int = 1000000000
)
```

AS

```
SELECT
    [Прізвище], [Ім'я], [ПоБатькові], Бригади.Назва AS
Бригада,
    Відділи.Назва AS Відділ, Рейси.ID AS РейсID,
    DATEDIFF(month, [ДатаНародження], GETDATE())/12 AS Вік,
    [ЗаробітнаПлата]
FROM Працівники
JOIN Відділи ON Працівники.ВідділиID = Відділи.ID
JOIN Бригади ON Працівники.БригадиID = Бригади.ID
JOIN Літаки ON Бригади.ID = Літаки.БригадиID
JOIN Рейси ON Літаки.ID = Рейси.ЛітакиID
WHERE (CASE WHEN @бригада IS NOT NULL THEN
        CASE WHEN (Бригади.Назва IN (SELECT Назва
FROM Бригади WHERE Назва LIKE @бригада)) THEN
            1
        ELSE
            0
        END
    ELSE
        1
    END) = 1
AND (CASE WHEN @відділ IS NOT NULL THEN
        CASE WHEN (Відділи.Назва IN (SELECT Назва
FROM Відділи WHERE Назва LIKE @відділ)) THEN
            1
        ELSE
            0
        END
    ELSE
        1
    END) = 1
```



```

AND (CASE WHEN @рейс IS NOT NULL THEN
        CASE WHEN (Рейси.ID = @рейс) THEN
            1
        ELSE
            0
        END
    ELSE
        1
    END) = 1
AND ЗаробітнаПлата BETWEEN @ЗПНижняМежа AND @ЗПВерхняМежа
AND DATEDIFF(month, ДатаНародження, GETDATE())/12 BETWEEN
@вікНижняМежа AND @вікВерхняМежа
AND Бригади.Назва IN (SELECT Бригади.Назва FROM Працівники
JOIN Бригади ON Працівники.БригадиID = Бригади.ID GROUP BY
Бригади.Назва HAVING AVG(ЗаробітнаПлата) BETWEEN
@середняЗПуБригадіНижняМежа AND @середняЗПуБригадіВерхняМежа)

ORDER BY РейсID, Бригада, Прізвище, [Ім'я], Побатькові

RETURN

```

3. ALTER PROCEDURE dbo.Query3

```
(
    @пройшли_неПройшли bit = NULL,
    @рікНижняМежа smallint = 0,
    @рікВерхняМежа smallint = 3000,
    @стать bit = NULL,
    @вікНижняМежа int = 0,
    @вікВерхняМежа int = 200,
    @ЗПНижняМежа int = 0,
    @ЗПВерхняМежа int = 1000000000
)
```

AS

```
    SELECT Прізвище, [Ім'я], Побатькові,
        DATEDIFF(month, [ДатаНародження], GETDATE())/12 AS Вік,
ЗаробітнаПлата
    FROM Працівники
    JOIN ЖурналМедоглядів ON Працівники.ID =
ЖурналМедоглядів.ПрацівникиID
    WHERE ВідділиID = (SELECT ID FROM Відділи WHERE Назва =
'пілоти')
    AND
    (
        CASE WHEN(@пройшли_неПройшли IS NOT NULL) THEN
            CASE WHEN(ЖурналМедоглядів.Результат =
@пройшли_неПройшли) THEN
                1
            ELSE
                0
            END
        ELSE
            1
        END
    ) = 1
    AND YEAR(ЖурналМедоглядів.Дата) BETWEEN @рікНижняМежа AND
@рікВерхняМежа
    AND
    (
        CASE WHEN @стать IS NOT NULL THEN
            CASE WHEN (Працівники.Стать = @стать) THEN
                1
            ELSE
                0
            END
        ELSE
            1
    )
```

```
        END
    ) = 1
    AND DATEDIFF(month, [ДатаНародження], GETDATE())/12 BETWEEN
@вікНижняМежа AND @вікВерхняМежа
    AND ЗаробітнаПлата BETWEEN @ЗПНижняМежа AND @ЗПВерхняМежа
    Group BY Прізвище, [Ім'я], Побатькові, ДатаНародження,
ЗаробітнаПлата
    ORDER BY Прізвище, [Ім'я], Побатькові, ДатаНародження,
ЗаробітнаПлата

    RETURN
```

4. ALTER PROCEDURE dbo.Query4

```
(
    @аеропорт nvarchar(20) = 'Чернівці',
    @датаЧасНижняМежа datetime = '12.17.1903 0:0',
    @датаЧасВерхняМежа datetime = '1.1.3000 0:0',
    @кількістьРейсівНижняМежа bigint = 0,
    @кількістьРейсівВерхняМежа bigint = 1000000
)
```

AS

```
SELECT Номер, ТипЛітаків.Тип, ДатаВипуску, КількістьМісць,
Бригади.Назва AS Бригада,
    (SELECT TOP 1 ДатаЧасПрильоту
     FROM Рейси
     JOIN Маршрути ON Рейси.МаршрутиID = Маршрути.ID
     WHERE @аеропорт = Маршрути.КінцевийПункт
     AND ДатаЧасПрильоту < @датаЧасВерхняМежа
     AND ДатаЧасПрильоту > @датаЧасНижняМежа
     AND ЛітакиID = Літаки.ID
     ORDER BY ДатаЧасПрильоту DESC
    ) AS ДатаЧасНатходження,
    (SELECT COUNT(*) FROM Рейси WHERE ЛітакиID = Літаки.ID) AS
КількістьЗдійсненихРейсів
FROM Літаки
JOIN ТипиЛітаків ON Літаки.ТипиЛітаківID = ТипиЛітаків.ID
JOIN Бригади ON Літаки.БригадиID = Бригади.ID
WHERE (CASE WHEN ((SELECT TOP 1 ДатаЧасПрильоту
                   FROM Рейси
                   JOIN Маршрути ON Рейси.МаршрутиID = Маршрути.ID
                   WHERE @аеропорт = Маршрути.КінцевийПункт
                   AND ДатаЧасПрильоту < @датаЧасВерхняМежа
                   AND ДатаЧасПрильоту > @датаЧасНижняМежа
                   AND ЛітакиID = Літаки.ID
                   ORDER BY ДатаЧасПрильоту DESC
                  ) IS NULL ) THEN 0 ELSE 1 END) = 1
AND (SELECT COUNT(*) FROM Рейси WHERE ЛітакиID = Літаки.ID)
BETWEEN @кількістьРейсівНижняМежа AND
@кількістьРейсівВерхняМежа

ORDER BY ДатаЧасНатходження, КількістьЗдійсненихРейсів DESC

RETURN
```

5. ALTER PROCEDURE dbo.Query5

```
(
    @чиПройшов bit = NULL,
    @датаНижняМежа date = '12.17.1903',
    @датаВерхняМежа date = '1.1.3000',
    @числоРемонтівНижняМежа int = 0,
    @числоРемонтівВерхняМежа int = 1000,
    @рейсівДоРемонтуНижняМежа int = 0,
    @рейсівДоРемонтуВерхняМежа int = 1000000,
    @вікНижняМежа int = 0,
    @вікВерхняМежа int = 1000
)
```

AS

```
SELECT Літаки.Номер, ТипЛітаків.Тип, CONVERT(nvarchar(12),
Літаки.ДатаВипуску, 104) AS ДатаВипуску, Літаки.КількістьМісць,
Бригади.Назва AS Бригада,
    DATEDIFF(month, ДатаВипуску, GETDATE())/12 AS Вік,
    (SELECT COUNT(*) FROM ЖурналТехоглядів WHERE ЛітакиID =
Літаки.ID AND РезультатТехогляду = 0) AS ЧислоРемонтів,
    (SELECT COUNT(*) FROM Рейси WHERE ЛітакиID = Літаки.ID AND
СтатусРейсуID = 0 AND ДатаЧасПрильоту <= (SELECT TOP 1
ДатаОгляду FROM ЖурналТехоглядів WHERE ЛітакиID = Літаки.ID AND
РезультатТехогляду = 0 ORDER BY ДатаОгляду)) AS РейсівДоРемонту

FROM ЖурналТехоглядів
JOIN Літаки ON ЖурналТехоглядів.ЛітакиID = Літаки.ID
JOIN ТипиЛітаків ON Літаки.ТипиЛітаківID = ТипиЛітаків.ID
JOIN Бригади ON Літаки.БригадиID = Бригади.ID

WHERE (CASE WHEN(@чиПройшов IS NOT NULL)THEN
    CASE WHEN(РезультатТехогляду = @чиПройшов)THEN
        1
    ELSE
        0
    END
    ELSE
        1
    END) = 1
    AND CONVERT(date, ДатаОгляду, 104) BETWEEN @датаНижняМежа
AND @датаВерхняМежа
    AND (SELECT COUNT(*) FROM ЖурналТехоглядів WHERE ЛітакиID =
Літаки.ID AND РезультатТехогляду = 0) BETWEEN
@числоРемонтівНижняМежа AND @числоРемонтівВерхняМежа
    AND (SELECT COUNT(*) FROM Рейси WHERE ЛітакиID = Літаки.ID
AND СтатусРейсуID = 0 AND ДатаЧасПрильоту <= (SELECT TOP 1
```

```
ДатаОгляду FROM ЖурналТехоглядів WHERE ЛітакиID = Літаки.ID AND  
РезультатТехогляду = 0 ORDER BY ДатаОгляду)) BETWEEN  
@рейсівДоРемонтуНижняМежа AND @рейсівДоРемонтуВерхняМежа  
AND DATEDIFF(month, ДатаВипуску, GETDATE())/12 BETWEEN  
@вікНижняМежа AND @вікВерхняМежа  
  
GROUP BY Номер, Тип, ДатаВипуску, КількістьМісць,  
Бригади.Назва, Літаки.ID  
ORDER BY РейсівДоРемонту, Вік  
  
RETURN
```

6. ALTER PROCEDURE dbo.Query6

```
(  
  @маршрут int = NULL,  
  @тривалістьПерельотуНижняМежа int = 0,  
  @тривалістьПерельотуВерхняМежа int = 1000,  
  @цінаКвиткаНижняМежа int = 0,  
  @цінаКвиткаВерхняМежа int = 1000000  
)
```

AS

```
  SELECT Літаки.Номер AS НомерЛітака, СтатусРейсу.Статус AS  
СтатусРейсу,  
  ТипиРейсів.Тип AS ТипРейсу, ЦінаКвитка,  
  DATEDIFF(hour, ДатаЧасВідльоту, ДатаЧасПрильоту) AS  
ТривалістьПерельоту  
FROM Рейси  
JOIN Літаки ON ЛітакиID = Літаки.ID  
JOIN СтатусРейсу ON СтатусРейсуID = СтатусРейсу.ID  
JOIN ТипиРейсів ON ТипиРейсівID = ТипиРейсів.ID  
WHERE (CASE WHEN(@маршрут IS NOT NULL)THEN  
  CASE WHEN(@маршрут = МаршрутиID)THEN 1 ELSE 0 END  
  ELSE 1 END) = 1  
  AND DATEDIFF(hour, ДатаЧасВідльоту, ДатаЧасПрильоту)  
BETWEEN @тривалістьПерельотуНижняМежа AND  
@тривалістьПерельотуВерхняМежа  
  AND ЦінаКвитка BETWEEN @цінаКвиткаНижняМежа AND  
@цінаКвиткаВерхняМежа  
  ORDER BY ТривалістьПерельоту, ЦінаКвитка  
  
RETURN
```

7. ALTER PROCEDURE dbo.Query7

```
(
    @напрямок nvarchar(30) = NULL,
    @маршрут int = NULL,
    @кількістьНевикористанихМісцьНижняМежа int = 0,
    @кількістьНевикористанихМісцьВерхняМежа int = 10000,
    @відсотокНевикористанихМісцьНижняМежа float = 0.0,
    @відсотокНевикористанихМісцьВерхняМежа float = 100.0
)
AS

SELECT Рейси.ID AS НомерРейсу, Літаки.Номер AS НомерЛітака,
ТипиРейсів.Тип AS ТипРейсу,
    (Літаки.КількістьМісць - (SELECT COUNT(*) FROM Квитки
WHERE РейсиID = Рейси.ID AND СтатусКвиткаID = 1) + (SELECT
COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND
СтатусКвиткаID = 0)) AS КількістьНевикористанихМісць,
    ((Літаки.КількістьМісць - (SELECT COUNT(*) FROM Квитки
WHERE РейсиID = Рейси.ID AND СтатусКвиткаID = 1) + (SELECT
COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND
СтатусКвиткаID = 0)) * 100.0 / Літаки.КількістьМісць) AS
ВідсотокНевикористанихМісць
FROM Рейси
JOIN Літаки ON ЛітакиID = Літаки.ID
JOIN ТипиРейсів ON ТипиРейсівID = ТипиРейсів.ID
JOIN Маршрути ON МаршрутиID = Маршрути.ID
JOIN СтатусРейсу ON СтатусРейсуID = СтатусРейсу.ID
WHERE СтатусРейсуID = (SELECT ID FROM СтатусРейсу WHERE
Статус = 'скасовано')
AND (CASE WHEN(@напрямок IS NOT NULL)THEN
CASE WHEN(@напрямок = Маршрути.КінцевийПункт)THEN
1 ELSE 0 END
ELSE 1 END) = 1
AND (CASE WHEN(@маршрут IS NOT NULL)THEN
CASE WHEN(@маршрут = МаршрутиID)THEN 1 ELSE 0 END
ELSE 1 END) = 1
AND (Літаки.КількістьМісць - (SELECT COUNT(*) FROM Квитки
WHERE РейсиID = Рейси.ID AND СтатусКвиткаID = 1) + (SELECT
COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND
СтатусКвиткаID = 0)) BETWEEN
@кількістьНевикористанихМісцьНижняМежа AND
@кількістьНевикористанихМісцьВерхняМежа
AND ((Літаки.КількістьМісць - (SELECT COUNT(*) FROM Квитки
WHERE РейсиID = Рейси.ID AND СтатусКвиткаID = 1) + (SELECT
COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND
СтатусКвиткаID = 0)) * 100.0 / Літаки.КількістьМісць) BETWEEN
```



```
@відсотокНевикористанихМісцьНижняМежа AND  
@відсотокНевикористанихМісцьВерхняМежа  
ORDER BY КількістьНевикористанихМісць,  
ВідсотокНевикористанихМісць
```

```
RETURN
```

8. ALTER PROCEDURE dbo.Query8

```
(  
    @причинаЗатримки nvarchar(50) = NULL,  
    @маршрут int = NULL,  
    @кількістьЗданихКвитківНижняМежа int = 0,  
    @кількістьЗданихКвитківВерхняМежа int = 10000  
)
```

AS

```
    SELECT Рейси.ID AS НомерРейсу, Літаки.Номер AS НомерЛітака,  
    ТипиРейсів.Тип AS ТипРейсу,  
    СтатусРейсу.Статус AS ПричинаЗатримки,  
    (SELECT COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND  
    СтатусКвиткаID = 0) AS КількістьЗданихКвитків  
    FROM Рейси  
    JOIN Літаки ON ЛітакиID = Літаки.ID  
    JOIN ТипиРейсів ON ТипиРейсівID = ТипиРейсів.ID  
    JOIN Маршрути ON МаршрутиID = Маршрути.ID  
    JOIN СтатусРейсу ON СтатусРейсуID = СтатусРейсу.ID  
    WHERE СтатусРейсуID IN (SELECT ID FROM СтатусРейсу WHERE  
    Статус LIKE '%затримано%')  
    AND (CASE WHEN(@причинаЗатримки IS NOT NULL)THEN  
        CASE WHEN(СтатусРейсуID IN (SELECT ID FROM  
    СтатусРейсу WHERE Статус LIKE '%' + @причинаЗатримки + '%'))THEN 1  
    ELSE 0 END  
        ELSE 1 END) = 1  
    AND (CASE WHEN(@маршрут IS NOT NULL)THEN  
        CASE WHEN(@маршрут = Рейси.МаршрутиID)THEN 1 ELSE  
    0 END  
        ELSE 1 END) = 1  
    AND (SELECT COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID  
    AND СтатусКвиткаID = 0) BETWEEN  
    @кількістьЗданихКвитківНижняМежа AND  
    @кількістьЗданихКвитківВерхняМежа  
    ORDER BY ПричинаЗатримки, КількістьЗданихКвитків  
  
    RETURN
```

9_1. ALTER PROCEDURE dbo.Query9_1

```
(
    @типЛітака nvarchar(20) = NULL,
    @тривалістьПерельотуНижняМежа int = 0,
    @тривалістьПерельотуВерхняМежа int = 1000,
    @цінаКвиткаНижняМежа int = 0,
    @цінаКвиткаВерхняМежа int = 1000000,
    @часВильотуНижняМежа time = '0:0:0',
    @часВильотуВерхняМежа time = '23:59:59'
)
```

AS

```
SELECT Рейси.ID AS НомерРейсу, Літаки.Номер AS НомерЛітака,
ТипиЛітаків.Тип AS ТипЛітака,
ТипиРейсів.Тип AS ТипРейсу,
СтатусРейсу.Статус AS СтатусРейсу,
DATEDIFF(hour, ДатаЧасВідльоту, ДатаЧасПрильоту) AS
ТривалістьПерельоту,
ЦінаКвитка,
CAST(DATENAME(hour, ДатаЧасВідльоту)+':'+DATENAME(minute,
ДатаЧасВідльоту)+':'+DATENAME(second, ДатаЧасВідльоту) AS time)
AS ЧасВильоту,
МаршрутиID AS НомерМаршруту
FROM Рейси
JOIN Літаки ON ЛітакиID = Літаки.ID
JOIN ТипиРейсів ON ТипиРейсівID = ТипиРейсів.ID
JOIN Маршрути ON МаршрутиID = Маршрути.ID
JOIN СтатусРейсу ON СтатусРейсуID = СтатусРейсу.ID
JOIN ТипиЛітаків ON ТипиЛітаківID = ТипиЛітаків.ID
WHERE (CASE WHEN(@типЛітака IS NOT NULL)THEN
CASE WHEN(ТипиЛітаків.Тип LIKE
'%' + @типЛітака + '%')THEN 1 ELSE 0 END
ELSE 1 END) = 1
AND DATEDIFF(hour, ДатаЧасВідльоту, ДатаЧасПрильоту)
BETWEEN @тривалістьПерельотуНижняМежа AND
@тривалістьПерельотуВерхняМежа
AND ЦінаКвитка BETWEEN @цінаКвиткаНижняМежа AND
@цінаКвиткаВерхняМежа
AND CAST(DATENAME(hour,
ДатаЧасВідльоту)+':'+DATENAME(minute,
ДатаЧасВідльоту)+':'+DATENAME(second, ДатаЧасВідльоту) AS time)
BETWEEN @часВильотуНижняМежа AND @часВильотуВерхняМежа
ORDER BY ТривалістьПерельоту, ЦінаКвитка, ЧасВильоту

RETURN
```

9_2. ALTER PROCEDURE dbo.Query9_2

```
(
    @типЛітака nvarchar(20) = NULL,
    @тривалістьПерельотуНижняМежа int = 0,
    @тривалістьПерельотуВерхняМежа int = 1000,
    @цінаКвиткаНижняМежа int = 0,
    @цінаКвиткаВерхняМежа int = 1000000,
    @часВильотуНижняМежа time = '0:0:0',
    @часВильотуВерхняМежа time = '23:59:59',
    @маршрут int = NULL
)
```

AS

```
SELECT Маршрут AS НомерМаршруту, AVG(ПроданоКвитків) AS
СередняКількістьПроданихКвитків,
    Маршрути.ПочатковийПункт, Маршрути.ПунктПересадки,
    Маршрути.КінцевийПункт
FROM
(
    SELECT Маршрути.ID AS Маршрут, Рейси.ID AS Рейс,
        CAST(COUNT(Рейси.ID) AS float) AS ПроданоКвитків
    FROM Квитки
    JOIN Рейси ON РейсиID = Рейси.ID
    JOIN Маршрути ON МаршрутиID = Маршрути.ID
    JOIN Літаки ON ЛітакиID = Літаки.ID
    JOIN ТипиЛітаків ON ТипиЛітаківID = ТипиЛітаків.ID
    WHERE Квитки.СтатусКвиткаID = 1
    AND (CASE WHEN(@маршрут IS NOT NULL)THEN
        CASE WHEN(Маршрути.ID = @маршрут)THEN 1 ELSE 0
    ELSE 1 END) = 1
    AND (CASE WHEN(@типЛітака IS NOT NULL)THEN
        CASE WHEN(ТипиЛітаків.Тип LIKE
        '%' + @типЛітака + '%')THEN 1 ELSE 0 END
        ELSE 1 END) = 1
    AND DATEDIFF(hour, ДатаЧасВідльоту, ДатаЧасПрильоту)
    BETWEEN @тривалістьПерельотуНижняМежа AND
    @тривалістьПерельотуВерхняМежа
    AND ЦінаКвитка BETWEEN @цінаКвиткаНижняМежа AND
    @цінаКвиткаВерхняМежа
    AND CAST(DATENAME(hour,
    ДатаЧасВідльоту) + ':' + DATENAME(minute,
    ДатаЧасВідльоту) + ':' + DATENAME(second, ДатаЧасВідльоту) AS time)
    BETWEEN @часВильотуНижняМежа AND @часВильотуВерхняМежа
    GROUP BY Рейси.ID, Маршрути.ID
) as Counts
```

```
JOIN Маршрути ON Маршрут = Маршрути.ID
GROUP BY Маршрут, ПочатковийПункт, ПунктПересадки,
КінцевийПункт

RETURN
```

10. ALTER PROCEDURE dbo.Query10

```
(  
    @типРейсу nvarchar(20) = NULL,  
    @напрямок nvarchar(30) = NULL,  
    @типЛітака nvarchar(20) = NULL  
)
```

AS

```
    SELECT Рейси.ID AS НомерРейсу, Літаки.Номер AS НомерЛітака,  
    ТипиРейсів.Тип AS ТипРейсу,  
    СтатусРейсу.Статус AS СтатусРейсу,  
    Маршрути.КінцевийПункт AS Напрямок,  
    ТипиЛітаків.Тип AS ТипЛітака  
FROM Рейси  
JOIN Літаки ON ЛітакиID = Літаки.ID  
JOIN ТипиРейсів ON ТипиРейсівID = ТипиРейсів.ID  
JOIN Маршрути ON МаршрутиID = Маршрути.ID  
JOIN СтатусРейсу ON СтатусРейсуID = СтатусРейсу.ID  
JOIN ТипиЛітаків ON ТипиЛітаківID = ТипиЛітаків.ID  
WHERE (CASE WHEN(@типРейсу IS NOT NULL)THEN  
        CASE WHEN(ТипиРейсів.Тип LIKE  
'%' + @типРейсу + '%') THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
    AND (CASE WHEN(@напрямок IS NOT NULL)THEN  
        CASE WHEN(Маршрути.КінцевийПункт LIKE  
'%' + @напрямок + '%') THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
    AND (CASE WHEN(@типЛітака IS NOT NULL)THEN  
        CASE WHEN(ТипиЛітаків.Тип LIKE  
'%' + @типЛітака + '%') THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
ORDER BY ТипРейсу, Напрямок, ТипЛітака  
  
RETURN
```

11. ALTER PROCEDURE dbo.Query11

```
(  
    @рейс bigint = NULL,  
    @деньВідльоту date = NULL,  
    @заКордон bit = NULL,  
    @наявністьБагажу bit = NULL,  
    @стать bit = NULL,  
    @вікНижняМежа int = 0,  
    @вікВерхняМежа int = 300  
)
```

AS

```
SELECT DISTINCT Рейси.ID AS НомерРейсу, ТипиРейсів.Тип AS  
ТипРейсу,  
    CONVERT(nvarchar(12), CAST(DATENAME(day,  
Рейси.ДатаЧасВідльоту) + '.' + DATENAME(month,  
Рейси.ДатаЧасВідльоту) + '.' + DATENAME(year,  
Рейси.ДатаЧасВідльоту) AS date), 104) AS ДеньВідльоту,  
    Пасажири.ПІБ AS Пасажир, Пасажири.Стать,  
    DATEDIFF(month, Пасажири.ДатаНародження, GETDATE())/12 AS  
Вік,  
    Квитки.НаявністьБагажу AS Багаж  
FROM Рейси  
JOIN ТипиРейсів ON Рейси.ТипиРейсівID = ТипиРейсів.ID  
JOIN Квитки ON Рейси.ID = Квитки.РейсиID  
JOIN Пасажири ON Квитки.ПасажириID = Пасажири.ID  
JOIN СтатусКвитка ON СтатусКвиткаID = СтатусКвитка.ID  
WHERE (CASE WHEN(@рейс IS NOT NULL)THEN  
    CASE WHEN(Рейси.ID = @рейс)THEN 1 ELSE 0 END  
    ELSE 1 END) = 1  
AND (CASE WHEN(@деньВідльоту IS NOT NULL)THEN  
    CASE WHEN(CONVERT(date, CAST(DATENAME(day,  
Рейси.ДатаЧасВідльоту) + '.' + DATENAME(month,  
Рейси.ДатаЧасВідльоту) + '.' + DATENAME(year,  
Рейси.ДатаЧасВідльоту) AS date), 104) = @деньВідльоту)THEN 1  
    ELSE 0 END  
    ELSE 1 END) = 1  
AND (CASE WHEN(@заКордон IS NOT NULL)THEN  
    CASE WHEN(@заКордон = (CASE WHEN(ТипиРейсів.Тип  
IN ('міжнародний','чартерний'))THEN CAST(1 AS bit) ELSE CAST(0  
AS bit) END))THEN 1 ELSE 0 END  
    ELSE 1 END) = 1  
AND (CASE WHEN(@наявністьБагажу IS NOT NULL)THEN  
    CASE WHEN(@наявністьБагажу =  
Квитки.НаявністьБагажу)THEN 1 ELSE 0 END  
    ELSE 1 END) = 1
```

```
        AND (CASE WHEN(@стать IS NOT NULL)THEN
                CASE WHEN(@стать = Пасажири.Стать)THEN 1 ELSE 0
END
                ELSE 1 END) = 1
        AND DATEDIFF(month, Пасажири.ДатаНародження, GETDATE())/12
        BETWEEN @вікНижняМежа AND @вікВерхняМежа
        AND (SELECT COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID
AND ПасажириID = Пасажири.ID AND СтатусКвиткаID = 0) = 0
        AND (SELECT COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID
AND ПасажириID = Пасажири.ID AND СтатусКвиткаID = 1) > 0
        AND СтатусКвитка.Статус = 'придбано'

RETURN
```


12. ALTER PROCEDURE dbo.Query12

```
(  
    @рейс bigint = NULL,  
    @день date = NULL,  
    @маршрут int = NULL,  
    @часВильотуНижняМежа time = '0:0:0',  
    @часВильотуВерхняМежа time = '23:59:59',  
    @цінаКвиткаНижняМежа int = 0,  
    @цінаКвиткаВерхняМежа int = 1000000  
)
```

AS

```
    SELECT Рейси.ID AS НомерРейсу, Рейси.МаршрутиID AS  
НомерМаршруту,  
        CONVERT(nvarchar(12), CAST(DATENAME(day,  
Рейси.ДатаЧасВідльоту)+'.'+DATENAME(month,  
Рейси.ДатаЧасВідльоту)+'.'+DATENAME(year,  
Рейси.ДатаЧасВідльоту) AS date), 104) AS ДеньВильоту,  
        CAST(DATENAME(hour,  
Рейси.ДатаЧасВідльоту)+':' +DATENAME(minute,  
Рейси.ДатаЧасВідльоту)+':' +DATENAME(second,  
Рейси.ДатаЧасВідльоту) AS time) AS ЧасВильоту,  
        Рейси.ЦінаКвитка,  
        (SELECT COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND  
СтатусКвиткаID = 2) AS ЗаброньованихМісць,  
        (Літаки.КількістьМісць - (SELECT COUNT(*) FROM Квитки  
WHERE РейсиID = Рейси.ID AND СтатусКвиткаID = 1) + (SELECT  
COUNT(*) FROM Квитки WHERE РейсиID = Рейси.ID AND  
СтатусКвиткаID = 0)) AS ВільнихМісць  
    FROM Рейси  
    JOIN Літаки ON ЛітакиID = Літаки.ID  
    WHERE (CASE WHEN(@рейс IS NOT NULL)THEN  
        CASE WHEN(Рейси.ID = @рейс)THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
    AND (CASE WHEN(@день IS NOT NULL)THEN  
        CASE WHEN(CONVERT(date, CAST(DATENAME(day,  
Рейси.ДатаЧасВідльоту)+'.'+DATENAME(month,  
Рейси.ДатаЧасВідльоту)+'.'+DATENAME(year,  
Рейси.ДатаЧасВідльоту) AS date), 104) = @день)THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
    AND (CASE WHEN(@маршрут IS NOT NULL)THEN  
        CASE WHEN(Рейси.МаршрутиID = @маршрут)THEN 1 ELSE  
0 END  
        ELSE 1 END) = 1  
    AND CAST(DATENAME(hour,  
Рейси.ДатаЧасВідльоту)+':' +DATENAME(minute,
```

```
Рейси.ДатаЧасВідльоту)+' ':' '+DATENAME(second,  
Рейси.ДатаЧасВідльоту) AS time) BETWEEN @часВильотуНижняМежа  
AND @часВильотуВерхняМежа  
AND Рейси.ЦінаКвитка BETWEEN @цінаКвиткаНижняМежа AND  
@цінаКвиткаВерхняМежа  
ORDER BY ЦінаКвитка, ЧасВильоту  
  
RETURN
```

13. ALTER PROCEDURE dbo.Query13

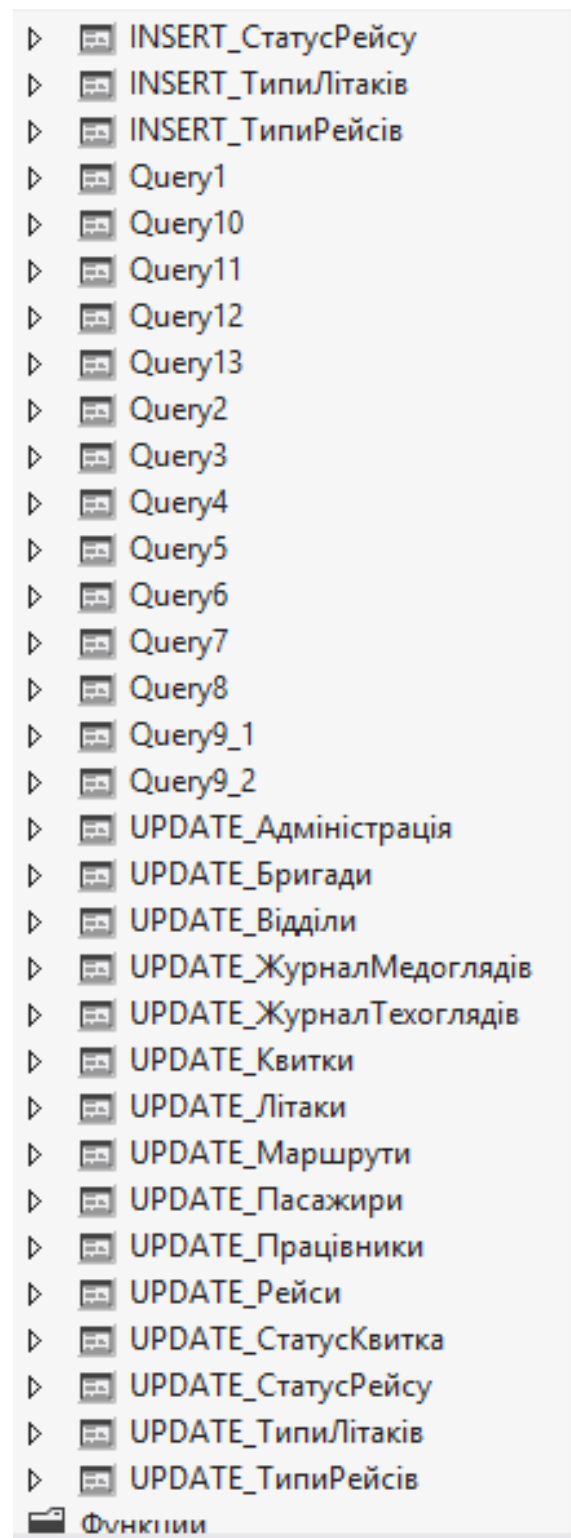
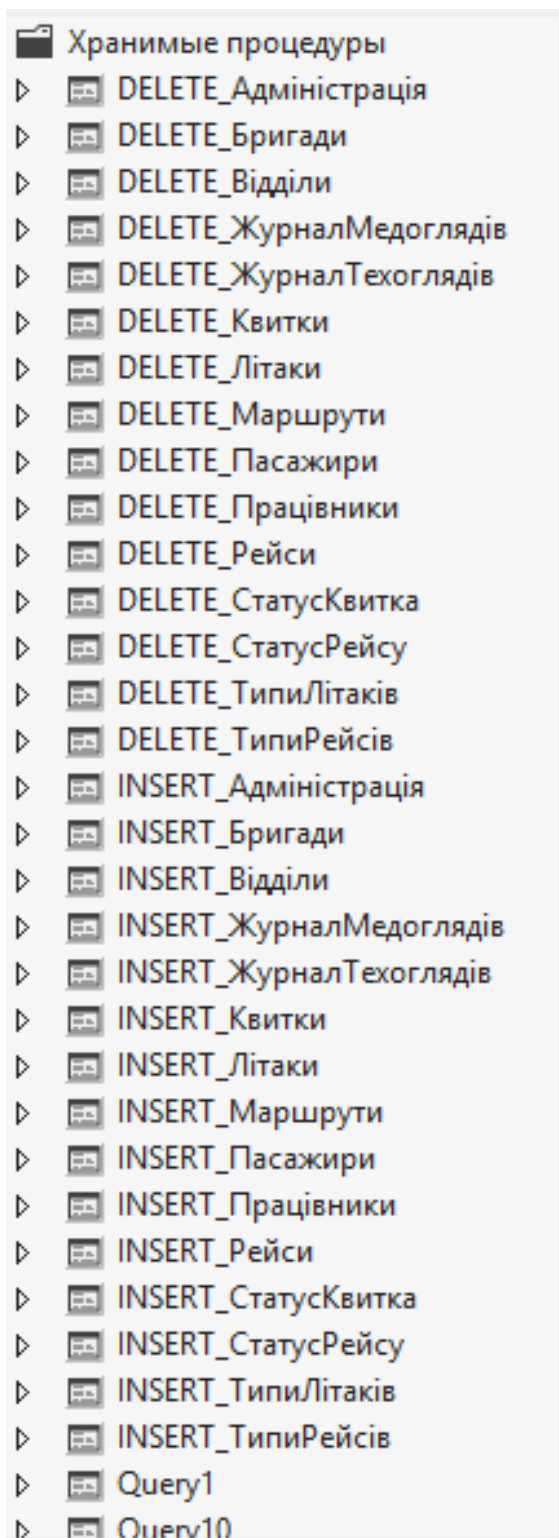
```
(  
    @рейс bigint = NULL,  
    @деньВильоту date = NULL,  
    @маршрут int = NULL,  
    @цінаКвиткаНижняМежа int = 0,  
    @цінаКвиткаВерхняМежа int = 1000000,  
    @вікНижняМежа int = 0,  
    @вікВерхняМежа int = 300,  
    @стать bit = NULL  
)
```

AS

```
SELECT COUNT(*) AS ЧислоЗданихКвитків  
FROM Квитки  
JOIN Рейси ON Квитки.РейсиID = Рейси.ID  
JOIN Пасажири ON Квитки.ПасажириID = Пасажири.ID  
WHERE Квитки.СтатусКвиткаID = 0  
AND (CASE WHEN(@рейс IS NOT NULL)THEN  
        CASE WHEN(Рейси.ID = @рейс)THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
AND (CASE WHEN(@деньВильоту IS NOT NULL)THEN  
        CASE WHEN(CONVERT(date, Рейси.ДатаЧасВідльоту,  
104) = @деньВильоту)THEN 1 ELSE 0 END  
        ELSE 1 END) = 1  
AND (CASE WHEN(@маршрут IS NOT NULL)THEN  
        CASE WHEN(Рейси.МаршрутиID = @маршрут)THEN 1 ELSE  
0 END  
        ELSE 1 END) = 1  
AND Рейси.ЦінаКвитка BETWEEN @цінаКвиткаНижняМежа AND  
@цінаКвиткаВерхняМежа  
AND DATEDIFF(month, Пасажири.ДатаНародження, GETDATE())/12  
BETWEEN @вікНижняМежа AND @вікВерхняМежа  
AND (CASE WHEN(@стать IS NOT NULL)THEN  
        CASE WHEN(Пасажири.Стать = @стать)THEN 1 ELSE 0  
END  
        ELSE 1 END) = 1  
  
RETURN
```

У результаті виконання створено 59 зберезуваних процедур:

- 14 - основні запити (9-ий розділено на два)
- 45 – реалізують додавання (INSERT), редагування (UPDATE) та видалення (DELETE) записів у кожній з 15 таблиць



Створюю програму, яка дозволить користувачу вводити параметри запитів, виконувати їх та виводити результати. Для її реалізації обираю мову **C#** з використанням платформи **.NetFramework** в середовищі програмування **Microsoft Visual Studio 2012 Ultimate**.

Інформаційна система Аеропорту

Query_1

1	Тільки начальники	Кількість дітей (нижня межа)	Заробітна плата (нижня межа)	Вік (нижня межа)	Стаж (нижня межа)	Стать
2		0	0	0	0	
3	Відділ	Кількість дітей (верхня межа)	Заробітна плата (верхня межа)	Вік (верхня межа)	Стаж (верхня межа)	
4		10	30000	100	100	

1

2

Елементи форми:

- **ListView** (властивість **Dock = Fill**) – для відображення результату запиту
- Контейнер **Panel** (властивість **Dock = Top**) – містить інші контейнери **Panel**, кожен з яких, в свою чергу, - набір елементів для задання параметрів відповідного запиту
- **MainMenu** – для перемикання **Panel** – ей з компонентами задання параметрів потрібного запиту

Фрагменти коду програми:

- Клік по 'Query_1' в **MainMenu** (Перемикання на першу панель)

```
private async void toolStripMenuItem2_Click(object sender, EventArgs e)
{
    hidePanels();
    showPanel("1");
    query1ToolStripMenuItem.Text = "Query_1";
    listView1.Clear();
    this.AcceptButton = button1;

    comboBox1.SelectedItem = "Bci";

    comboBox2.Items.Clear();
    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT Назва FROM Відділи", sqlConnection);
    try
    {
        sqlReader = await command.ExecuteReaderAsync();
        while (await sqlReader.ReadAsync())
        {
            comboBox2.Items.Add(Convert.ToString(sqlReader["Назва"]));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}
```

- Клік по 'button1' на першій панелі (Виконання першого запиту)

```
private async void button1_Click(object sender, EventArgs e)
{
    listView1.Clear();

    listView1.Columns.Add("Прізвище");
    listView1.Columns.Add("Ім'я");
    listView1.Columns.Add("Побатькові");
    listView1.Columns.Add("Відділ");
    listView1.Columns.Add("КількістьДітей");
    listView1.Columns.Add("ЗаробітнаПлата");
    listView1.Columns.Add("Вік");
    listView1.Columns.Add("Стаж");

    SqlDataReader sqlReader = null;

    SqlCommand command = new SqlCommand("Query1", sqlConnection);
    command.CommandType = CommandType.StoredProcedure;

    if(comboBox1.Text == "начальники")
        command.Parameters.AddWithValue("@начальники", "начальники");
    else
        command.Parameters.AddWithValue("@начальники", null);

    if (comboBox2.Items.IndexOf(comboBox2.Text) >= 0)
        command.Parameters.AddWithValue("@відділ", comboBox2.Text);
    else
        command.Parameters.AddWithValue("@відділ", null);

    if(comboBox3.Text == "жін")
        command.Parameters.AddWithValue("@стать", false);
    else if(comboBox3.Text == "чол")
        command.Parameters.AddWithValue("@стать", true);
    else
        command.Parameters.AddWithValue("@стать", null);

    command.Parameters.AddWithValue("@кількістьДітейНижняМежа", maskedTextBox1.Text);
    command.Parameters.AddWithValue("@кількістьДітейВерхняМежа", maskedTextBox2.Text);

    command.Parameters.AddWithValue("@ЗПНижняМежа", maskedTextBox4.Text);
    command.Parameters.AddWithValue("@ЗПВерхняМежа", maskedTextBox3.Text);

    command.Parameters.AddWithValue("@ВікНижняМежа", maskedTextBox6.Text);
    command.Parameters.AddWithValue("@ВікВерхняМежа", maskedTextBox5.Text);

    command.Parameters.AddWithValue("@СтажНижняМежа", maskedTextBox8.Text);
    command.Parameters.AddWithValue("@СтажВерхняМежа", maskedTextBox7.Text);

    try
    {
        sqlReader = await command.ExecuteReaderAsync();

        while (await sqlReader.ReadAsync())
        {
            ListViewItem item = new ListViewItem(new string[] {
                Convert.ToString(sqlReader["Прізвище"]),
                Convert.ToString(sqlReader["Ім'я"]),
                Convert.ToString(sqlReader["Побатькові"]),
                Convert.ToString(sqlReader["Відділ"]),
                Convert.ToString(sqlReader["КількістьДітей"]),
                Convert.ToString(sqlReader["ЗаробітнаПлата"]),
                Convert.ToString(sqlReader["Вік"]),
                Convert.ToString(sqlReader["Стаж"])
            });

            listView1.Items.Add(item);
        }

        listView1.Items.Add("");
        listView1.Items.Add("");
        ListViewItem item1 = new ListViewItem(new string[] { "Всього:", (listView1.Items.Count - 2).ToString() });
        listView1.Items.Add(item1);

        listView1.AutoResizeColumns(ColumnHeaderAutoResizeStyle.HeaderSize);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}
```

- Поля класу форми та додаткові методи

```
public partial class Form1 : Form
{
    SqlConnection sqlConnection;

    CultureInfo provider = CultureInfo.InvariantCulture;

    public void hidePanels()
    {
        panel1.Visible = false;
        panel2.Visible = false;
        panel3.Visible = false;
        panel4.Visible = false;
        panel5.Visible = false;
        panel6.Visible = false;
        panel7.Visible = false;
        panel8.Visible = false;
        panel9_1.Visible = false;
        panel9_2.Visible = false;
        panel10.Visible = false;
        panel11.Visible = false;
        panel12.Visible = false;
        panel13.Visible = false;
    }

    public void showPanel(string panel)
    {
        switch (panel)
        {
            case "1":
            {
                panel1.Visible = true;
                break;
            }
            case "2":
            {
                panel2.Visible = true;
                break;
            }
            case "3":
            {
                panel3.Visible = true;
                break;
            }
            case "4":
            {
```

- Закриття зв'язку з базою даних при закритті форми

```
private void Form1_FormClosing(object sender, FormClosingEventArgs e)
{
    if (sqlConnection != null && sqlConnection.State != ConnectionState.Closed)
        sqlConnection.Close();
}
```


- Встановлення зв'язку з базою та початкові налаштування при завантаженні форми

```
private async void Form1_Load(object sender, EventArgs e)
{
    string connectionString = @"Data Source=COMPUTER;Initial Catalog=Airport;Integrated Security=True";

    sqlConnection = new SqlConnection(connectionString);

    await sqlConnection.OpenAsync();

    hidePanels();
    showPanel("1");
    listView1.Clear();
    this.AcceptButton = button1;

    panelTop.VerticalScroll.Maximum = 0;
    panelTop.AutoScroll = false;
    panelTop.VerticalScroll.Visible = false;
    panelTop.AutoScroll = true;

    comboBox2.Items.Clear();
    SqlDataReader sqlReader = null;
    SqlCommand command = new SqlCommand("SELECT Назва FROM Відділи", sqlConnection);
    try
    {
        sqlReader = await command.ExecuteReaderAsync();
        while (await sqlReader.ReadAsync())
        {
            comboBox2.Items.Add(Convert.ToString(sqlReader["Назва"]));
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message.ToString(), ex.Source.ToString(), MessageBoxButtons.OK, MessageBoxIcon.Error);
    }
    finally
    {
        if (sqlReader != null)
            sqlReader.Close();
    }
}
```

1

2

Інформаційна система Аеропорту

Query_2

Виконати

Бригада
Колобок ▾

Середня ЗП у бригаді (нижня межа)
0 _____

Заробітна плата (нижня межа)
0 _____

Вік (нижня межа)
0 _____

Рейс
7 ▾

Відділ
_____ ▾

Середня ЗП у бригаді (верхня межа)
30000 _____

Заробітна плата (верхня межа)
30000 _____

Вік (верхня межа)
35 _____

Прізвище	Ім'я	ПоБатькові	Бригада	Відділ	РейсID	Вік	ЗаробітнаПлата
Варава	Петро	Васильович	Колобок	довідкова служба	7	25	1000
Глина	Михайло	Васильович	Колобок	обслуговуючий персонал	7	18	4000
Кошман	Анатолій	Володимирович	Колобок	диспетчери	7	31	9000
Плішка	Інна	Ярославівна	Колобок	техніки	7	26	6900
Всього:		4					

3

Інформаційна система Аеропорту

Query_3

Виконати

Результат:

Рік (нижня межа):

Заробітна плата (нижня межа):

Вік (нижня межа):

Стать:

Рік (верхня межа):

Заробітна плата (верхня межа):

Вік (верхня межа):

Прізвище	Ім'я	ПоБатькові	Вік	ЗаробітнаПлата
Сира	Анна	Дмитрівна	44	25000
Соломій	Інокентій	Павлович	26	15000
Всього: 2				

4

Інформаційна система Аеропорту

Query_4

Виконати

Аеропорт:

Дата й час (нижня межа):

Кількість рейсів (нижня межа):

Дата й час (верхня межа):

Кількість рейсів (верхня межа):

Номер	Тип	ДатаВипуску	КількістьМісць	Бригада	ДатаЧасНатходження	КількістьЗдійсненихРейсів
K435AD43	пасажирський	01.05.2010 0:00:00	300	Карась	02.01.2018 4:00:00	5
RT5435435	військовий	15.10.2007 0:00:00	50	Колобок	20.03.2018 13:00:00	2
LFDJ32552	пасажирський	01.08.2008 0:00:00	250	Баклажан	04.04.2018 13:00:00	4
Всього: 3						

5

Інформаційна система Аеропорту

Query_5

Результат:

Дата (нижня межа): Число ремонтів (нижня межа): Рейсів до ремонту (нижня межа): Вік (нижня межа):

Дата (верхня межа): Число ремонтів (верхня межа): Рейсів до ремонту (верхня межа): Вік (верхня межа):

Номер	Тип	ДатаВипуску	КількістьМісць	Бригада	Вік	ЧислоРемонтів	РейсівДоРемонту
LFDJ32552	пасажирський	01.08.2008	250	Баклажан	9	1	2
K435AD43	пасажирський	01.05.2010	300	Карась	7	1	3
Всього:	2						

6

Інформаційна система Аеропорту

Query_6

Маршрут:

Тривалість перельоту (нижня межа): Ціна квитка (нижня межа):

Тривалість перельоту (верхня межа): Ціна квитка (верхня межа):

НомерПітака	СтатусРейсу	ТипРейсу	ЦінаКвитка	ТривалістьПерельоту
K435AD43	виконано	внутрішній	1000	2
K435AD43	затримано через погодні умови	внутрішній	1700	2
Всього:	2			

7

Інформаційна система Аеропорту

Query_7

Виконати

Маршрут: 2 Париж/П'єв'єв/Чернівці

Напрямок: Чернівці

Кількість невикористаних місць (нижня межа): 0

Кількість невикористаних місць (верхня межа): 1000

Відсоток невикористаних місць (нижня межа): 0

Відсоток невикористаних місць (верхня межа): 100

НомерРейсу	НомерЛітака	ТипРейсу	КількістьНевикористанихМісць	ВідсотокНевикористанихМісць
12	RT5435435	міжнародний	49	98,00000000
Всього:	1			

8

Інформаційна система Аеропорту

Query_8

Виконати

Причина затримки:

Маршрут:

Кількість зданих квитків (нижня межа): 0

Кількість зданих квитків (верхня межа): 1000

НомерРейсу	НомерЛітака	ТипРейсу	ПричинаЗатримки	КількістьЗданихКвитків
13	LFDJ32552	міжнародний	затримано з технічних причин	0
8	K435AD43	внутрішній	затримано через погодні умови	1
Всього:	2			

9_1

Інформаційна система Аеропорту

Query_9_1

Виконати

Тип літака:

Тривалість перельоту (нижня межа):

Ціна квитка (нижня межа):

Час вильоту (нижня межа):

Тривалість перельоту (верхня межа):

Ціна квитка (верхня межа):

Час вильоту (верхня межа):

НомерРейсу	НомерЛітака	ТипЛітака	ТипРейсу	СтатусРейсу	ТривалістьПерельоту	ЦінаКвитка	ЧасВильоту	НомерМаршруту
13	LFDJ32552	пасажирський	міжнародний	затримано з технічних причин	1	1700	12:00:00	4
0	K435AD43	пасажирський	внутрішній	виконано	2	1000	15:00:00	0
9	K435AD43	пасажирський	внутрішній	виконано	2	1100	15:00:00	2
5	K435AD43	пасажирський	внутрішній	виконано	2	1200	15:00:00	2
8	K435AD43	пасажирський	внутрішній	затримано через погодні умови	2	1700	15:00:00	0
2	LFDJ32552	пасажирський	внутрішній	виконано	5	1500	15:00:00	1
10	LFDJ32552	пасажирський	внутрішній	виконано	29	900	00:00:00	1
6	LFDJ32552	пасажирський	внутрішній	виконано	29	1000	15:00:00	3
Всього:	8							

9_2

Інформаційна система Аеропорту

Query_9_2

Виконати

Тип літака:

Маршрут:

Тривалість перельоту (нижня межа):

Ціна квитка (нижня межа):

Час вильоту (нижня межа):

Тривалість перельоту (верхня межа):

Ціна квитка (верхня межа):

Час вильоту (верхня межа):

НомерМаршруту	СередняКількістьПроданихКвитків	ПочатковийПункт	ПунктПересадки	КінцевийПункт
0	3	Чернівці	Львів	Париж
2	2	Париж	Львів	Чернівці
Всього:	2			

11

Інформаційна система Аеропорту

Query_11

Виконати

Рейс

За кордон

Стать

Вік (нижня межа)

Вік (верхня межа)

День відльоту

Наявність багажу

НомерРейсу	ТипРейсу	ДеньВідльоту	Пасажири	Стать	Вік	Багаж
0	внутрішній	15.10.2017	Гумільов Валерій Полікарпович	True	32	True
7	спеціальний	20.03.2018	Жук Марина Сергіївна	False	28	True
8	внутрішній	21.10.2017	Жук Марина Сергіївна	False	28	True
12	міжнародний	20.03.2018	Гумільов Валерій Полікарпович	True	32	True
Всього:	4					

12

✈ Інформаційна система Аеропорту

Query_12

Виконати

Рейс

Маршрут

Час вильоту (нижня межа)

Ціна квитка (нижня межа)

День

Час вильоту (верхня межа)

Ціна квитка (верхня межа)

00:00:00

0

23:59:59

10000

НомерРейсу	НомерМаршруту	ДеньВильоту	ЧасВильоту	ЦінаКвитка	ЗаброньованихМісць	ВільнихМісць
10	1	04.01.2018	00:00:00	900	0	250
0	0	15.10.2017	15:00:00	1000	0	297
6	3	15.10.2017	15:00:00	1000	0	250
9	2	22.10.2017	15:00:00	1100	0	300
5	2	16.10.2017	15:00:00	1200	0	300
2	1	15.10.2017	15:00:00	1500	0	250
13	4	04.04.2018	12:00:00	1700	0	250
8	0	21.10.2017	15:00:00	1700	0	299
12	2	20.03.2018	10:00:00	2000	0	49
Всього:		9				

13

✈ Інформаційна система Аеропорту

Query_13

Виконати

Рейс

Маршрут

Вік (нижня межа)

Ціна квитка (нижня межа)

День вильоту

Стать

Вік (верхня межа)

Ціна квитка (верхня межа)

0

0

100

10000

Число ЗданихКвитків

1

Висновки: У результаті виконання поставленого завдання я навчився створювати базу даних у середовищі **Microsoft SQL Server Management Studio**, проектувати її таблиці та зв'язки між ними, створювати збережені процедури для маніпулювання таблицями та їх вмістом, у середовищі **Microsoft Visual Studio**. Також я навчився звертатися до бази даних на сервері SQL через засоби .NetFramework, та розробив програму, яка дозволяє користувачу задавати параметри запитів та отримувати результат їх виконання через простий інтерфейс взаємодії.

