НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ СІКОРСЬКОГО КАФЕДРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ КОМП’ЮТЕРНИХ СИСТЕМ

**ПОЯСНЮВАЛЬНА ЗАПИСКА**

**до курсової роботи**

з дисципліни

**«ОСНОВИ ПРОГРАМУВАННЯ»**

на тему

«**Інформаційна система нарахувань заробітної плати в  бухгалтерії підприємства**»

Викладач В.В. Хайдуров

Студентка Кирильчук Олександра Артурівна

Група КП-11

Варіант 37

Київ – 2022

***Завдання до курсової роботи***

*Варіант 37.* Інформаційна система нарахувань заробітної плати в  бухгалтерії підприємства

Зарплата нараховується працівникам залежно від встановленого окладу.  У період хвороби працівнику нараховується % зарплати, який залежить від  стажу робітника. Якщо стаж до 5 років – 50%, до 10 років – 75%, інакше 90%.  Із загальної суми зарплати відраховується прибутковий податок, який  залежить від суми зарплати. Якщо сума до 20 000 грн – 15%, інакше 17%.

Дані співробітників: номер особистої картки; П.І.Б.; посада; оклад;  сімейний стан і кількість дітей; дані про невихід на роботу через хворобу (дати  захворювання та одужання) тощо.

Працівникам нараховуються премії та надбавки. Премії нараховують  один раз, надбавки встановлюються на певний період часу. Підприємство переховує в бюджет податок на фонд заробітної плати в розмірі  40% від нарахованих сум за поточний місяць.

Вихідні дані:

− формування списків працівників з розміром зарплатні за місяць;

− відомостей про працівників з лікарняним;

− виведення загальної суми нарахувань, премій та надбавок за місяць;

− виведення загальної суми перерахованих за місяць податків.

Функціонал:

− ведення обліку та видача відомостей виплат;

− забезпечення перегляду та друку всіх вихідних даних;

− уведення даних про лікарняні листи;

− уведення даних про премії та надбавки.

Звіт

[*Завдання до курсової роботи* 2](#_Toc106873430)

[ВСТУП 4](#_Toc106873431)

[*Актуальність досліджень* 4](#_Toc106873432)

[*Мета роботи* 5](#_Toc106873433)

[*Завдання дослідження* 6](#_Toc106873434)

[*Порядок виконання завдання* 6](#_Toc106873435)

[*Засоби реалізації завдань проектування* 7](#_Toc106873436)

[1. Необхідність створення  інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства 8](#_Toc106873437)

[2. Функціональне моделювання інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства з використанням DFD-діаграм 9](#_Toc106873438)

[3. Концептуальна модель бази даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства 12](#_Toc106873439)

[4. Фізична модель бази даних системи нарахувань заробітної плати в  бухгалтерії підприємства 14](#_Toc106873440)

[5. Опис реалізації завдань відповідно до обраного варіанту. Створення та структура таблиць бази даних інформаційної  системи нарахувань заробітної плати в бухгалтерії підприємства 19](#_Toc106873441)

[5.1 Формування запитів до створеної бази даних системи нарахувань заробітної плати в бухгалтерії підприємства 21](#_Toc106873442)

[5.2. Розроблення інтерфейсної частини для роботи зі створеною  базою даних системи нарахувань заробітної плати в бухгалтерії підприємства 28](#_Toc106873444)

[6. Тестування розробленої бази даних інформаційної системи нарахувань заробітної плати в бухгалтерії підприємства 30](#_Toc106873445)

[6.1. Тестування запитів до бази даних 30](#_Toc106873446)

[6.2. Тестування представлень та процедур до бази даних 33](#_Toc106873447)

[6.3. Тестування додавання та виделення даних про працівників за ID 37](#_Toc106873448)

[*Висновки* 38](#_Toc106873449)

[Додаток А 42](#_Toc106873450)

[Додаток Б 47](#_Toc106873451)

[Додаток В 50](#_Toc106873452)

**ВСТУП**

## *Актуальність досліджень*

Функціонування будь-якого підприємства має на меті створення прибутку. Створенням прибутку займаються працівники підприємства, які за свою роботу отримують фінансову, або інші різновиди, винагороду. Баталії на полі боя виграють кращі воїнські формування, баталії на ринках товарів та послуг виграють найкращі колективи працівників. Створення в колективі працівників атмосфери відповідності заробітної плати дійсному внеску членів колективу, рівень їхньої справедливості безпосередньо впливають на лояльність колективу до керівництва, на дотримання колективом ідеології та стратегічних цілей підприємства, на здатність та готовність робітників до самовідданої праці. Якщо ж працівники будуть сприймати розподіл грошової винагороди, як несправедливий, то ніщо не завадить їм саботувати виконання своїх обов’язків, покладених на них посадовими інструкціями та трудовими договорами. І цей саботаж дуже швидко розповсюдиться по всьому підприємству, наче гниль по дереву, і буде так само згубним для самого виживання комерційної структури в умовах конкурентного тиску на ринках. Неможливо переоцінити той рівень фінансового внеску найкращих працівників в прибуток підприємства, тому найперша задача керівництва – це розробити та організувати нарахування та отримання заробітної плати в такий спосіб, щоб її рівень максимально відповідав безпосередньому внеску конкретного працівника. А це має включати цілу низку факторів, таких, як кількість робочих днів в місяці; кількість робочих годин в конкретному дні та за місяць загалом; святкові та вихідні дні; пропуски по непрацездатності з різних причин; нарахування за відрядження; надбавки та премії за виконання плану, або додаткових завдань; додаткові оплати за роботу у вихідні, понаднормово та в нічний час; авторські, патентні виплати та виплати за винаходи; пайові відсотки і таке інше. Відповідно, чим більше керівництво підприємства за допомогою своїх бухгалтерів наблизиться до ідеального розподілу винагороди своїм працівникам, тим більше прибутку в свою чергу ці робітники забезпечать власникам та акціонерам компанії. Тобто, досконала організація процесу нарахування та виплати грошового забезпечення колективу – це безпосередня умова існування підприємства. Якщо врахувати всі наведені фактори та оцінити механічну кількість роботи, яку потрібно виконувати відділу бухгалтерії на великому підприємстві, то безсумнівно сама ця праця викликає в свою чергу необхідність утримання значного штату бухгалтерів первинної ланки. Проте, це означає значне збільшення витрат на цю роботу та вплив людського фактору, тобто, можливості помилки через значну кількість даних для обробки. Комп’ютеризація цих процесів дозволяє скоротити склад персоналу відділу бухгалтерії, автоматизувати процес нарахування заробітної платні, уможливити внесення первинних даних в систему нарахування безпосередньо самими робітниками. Всі знають, що існує лише два способи збільшити прибуток - це збільшити надходження, або зменшити витрати. В умовах сучасного рівня конкуренції на насичених ринках, особливо в такий період, який переживає наша країна, а саме стан затяжної війни, збільшення надходжень майже неможлива в будь-якій галузі. Стабілізація фінансового потоку на довоєнному рівні була б найкращим сценарієм для більшості підприємств. Відповідно, здатність компаній до скорочення витрат буде визначальним фактором, який забезпечить виживання колективів в найближчі роки. Тому залучення комп’ютерних технологій, автоматизація процесів, скорочення ручної праці та кількості бухгалтерів є єдино вірною та можливою тактикою будь-якого підприємства.

## *Мета роботи*

Метою даної роботи є створення функціонуючої системи нарахування та виплати заробітної плати та заохочень для працівників комерційного підприємства.

## *Завдання дослідження*

Завдання дослідження полягає в розробці та створенні інформаційної системи, яка має виконувати функцію нарахування грошової винагороди робітників комерційного підприємства. Ця система передбачає внесення та обробку даних стосовно анкетних даних робітників, в тому числі сімейного стану та кількості дітей; займаних посад та посадових окладів. Має бути розроблений інтерфейс для самостійного дистанційного табелювання працівників, а саме внесення інформації про невихід на роботу через захворювання. На підставі анкетних даних, які дозволяють розрахувати робочий стаж працівників, та інформації про невихід на роботу система має проводити розрахунок виплати лікарняних, а також щомісячні надбавки за безперервну роботу та преміювання за виконання плану за рік. Внесення актуальної законодавчої бази про оподаткування та нарахування грошової винагороди мають забезпечити автоматичний розрахунок податків та зборів на заробітну платню та їхнє автоматичне списання з рахунків підприємства, формування податкової звітності та її передавання в режимі онлайн до регіонального податкового органу.

## *Порядок виконання завдання*

Робота є структурованою за розділами.  Перед проектуванням інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємствапроведено аналіз відповідної предметної області.  Визначивши основні аспекти роботи розглядуваних підприємств,  визначено основні сутності, які використовуються при створенні відповідної  бази даних обліку заробітної плати в підприємстві. Проведено аналіз сутностей та  визначено між ними логічні зв’язки для повноцінної роботи бази даних, яка є  основним структурним і функціональним елементом інформаційної системи в  цілому. База даних наповнена відповідними даними, які відображають  дані про співробітників, а саме: номер особистої картки, П.І.Б., посада, оклад,  сімейний стан і кількість дітей, дані про невихід на роботу через хворобу (дати  захворювання та одужання), а також дані про податки, премії та надбавки. Формування таких даних відображено у кількох  таблицях. Будь-які дві зв’язані таблиці мають зв’язок «один до одного».  Після формування повноти даних у створеній базі, створено відповідні  тригери для забезпечення цілісності даних при роботі з даними у самій базі.  Наступним кроком є створення різних типів запитів до даної бази, які  відображають її повноцінну роботу. Після створення бази даних створюється  інтерфейс, за допомогою якого будь-який користувач, зокрема, оператор  мережі кінотеатрів має можливість виконувати обробку даних (каскадні зміни  у базі, а також редагування даних «на ходу»). Отже, створена база даних та  графічний інтерфейс, що призначений для неї і є основними функціональними  елементами інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства.

## *Засоби реалізації завдань проектування*

У роботі використано два  основних засоби для розробки інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства. Це Microsoft SQL Server Management Studio 18, за  допомогою якого виконується розробка відповідної бази інформаційної системи нарахувань заробітної плати, а також Microsoft Visual Studio 2019, за допомогою  якого створено програмний інтерфейс мовою програмування C# (WPF), який  дає змогу максимально спросити роботу користувача даною інформаційною системою, а також отримувати необхідні дані щодо кінцевої заробітної плати динамічно у часі.

# Необхідність створення  інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства

Будь-яка інформаційні система завжди працює з різними даними, які  повинна ефективно обробляти. До обробки даних можна віднести додавання  даних, видалення, оновлення даних, створення запитів звітів тощо. При цьому  створення такої інформаційної системи передбачає навики та уміння  проектування бази даних та зручного й інтуїтивного інтерфейсу користувача,  через який користувач інформаційної системи взаємодіє із базою даних.  Завданням курсової роботи є створення інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства.

Очевидним є той факт, що найперша задача керівництва – це розробити та організувати нарахування та отримання заробітної плати в такий спосіб, щоб її рівень максимально відповідав безпосередньому внеску конкретного працівника.

Виходячи з аналізу предметної області слід зазначити, що розроблювана  інформаційна система повинна мати гнучкий функціонал, зокрема повинна мати можливість:

* формування списків працівників з розміром зарплатні за місяць;
* відомостей про працівників з лікарняним;
* виведення загальної суми нарахувань, премій та надбавок за місяць;
* виведення загальної суми перерахованих за місяць податків;
* ведення обліку та видача відомостей виплат;
* забезпечення перегляду та друку всіх вихідних даних;
* уведення даних про лікарняні листи;
* уведення даних про премії та надбавки.
* мати зручний користувацький інтерфейс.

# 2. Функціональне моделювання інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства з використанням DFD-діаграм

*DFD* – загальноприйняте скорочення від англ. data flow diagrams – діаграми потоків даних. Так називається методологія графічного структурного  аналізу, що описує зовнішні по відношенню до системи джерела і адресати  даних, логічні функції, потоки даних і сховища даних, до яких здійснюється  доступ. Діаграма потоків даних (data flow diagram, *DFD*) – один з основних  інструментів структурного аналізу і проектування інформаційних систем, що  існували до широкого поширення UML [1; 5; 6].

Діаграми потоків даних використовуються для опису руху документів і обробки інформації як доповнення до IDEF0. На відміну від IDEF0, де система розглядається як пов'язані між собою роботи і стрілки представляють собою жорсткі взаємозв'язку, стрілки в DFD показують лише те, як об'єкти (включаючи дані) рухаються від однієї роботи до іншої. DFD відображає функціональні залежності значень, що обчислюються в системі, включаючи вхідні значення, вихідні значення і внутрішні сховища даних. ***DFD***- Це граф, на якому показано рух значень даних від їх джерел через перетворюють їх процеси до їх споживачам в інших об'єктах.

DFD містить процеси, які перетворюють дані, потоки даних, які переносять дані, активні об'єкти, які виробляють і споживають дані, і сховища даних, які пасивно зберігають дані.

На рис. 2.1 показано діаграму потоків даних при роботі з даними, які  містяться у базі даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства.

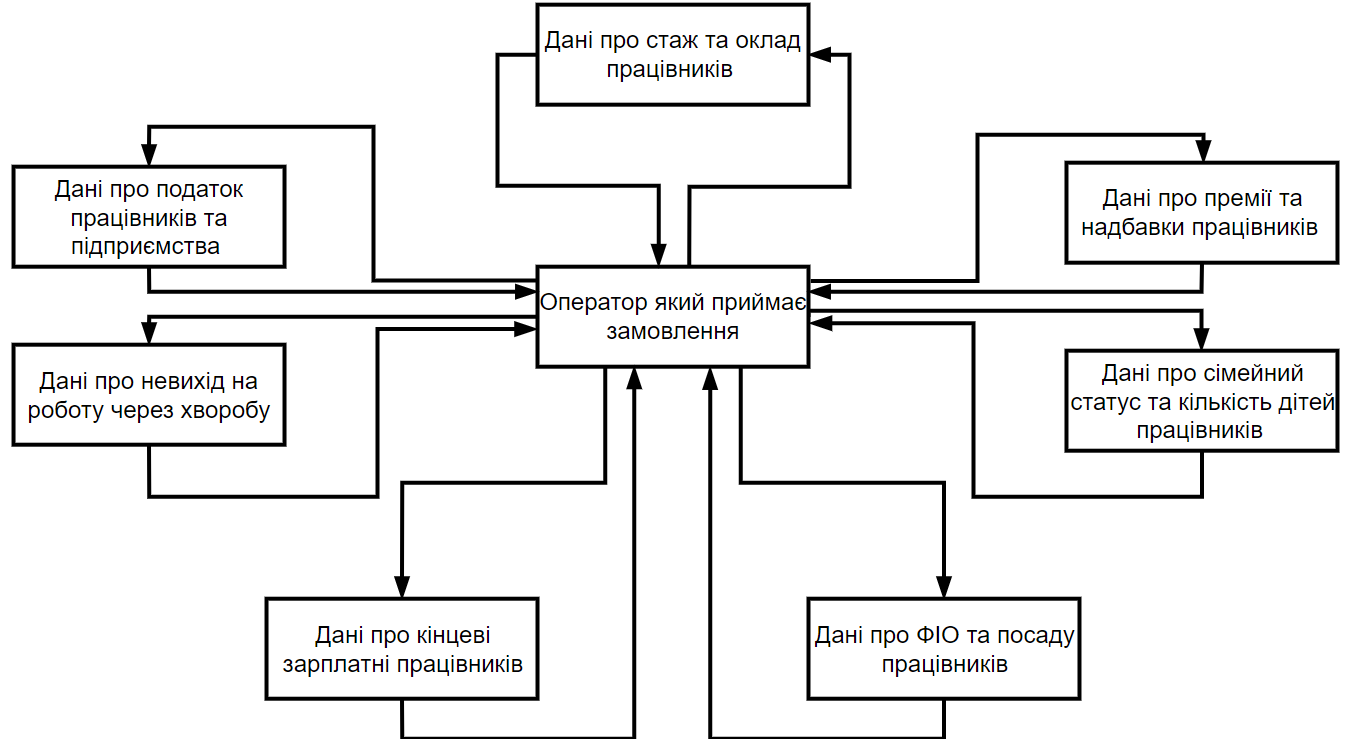


Рис.2.1. Діаграма потоків даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства

На рис. 2.2 показано діаграму прецедентів при роботі з програмним  засобом, що є проектованою інформаційною системою нарахувань заробітної плати в  бухгалтерії підприємства. У роботі програмно реалізовано усі блоки (рис. 2.2)  для формування роботиінформаційної системи.

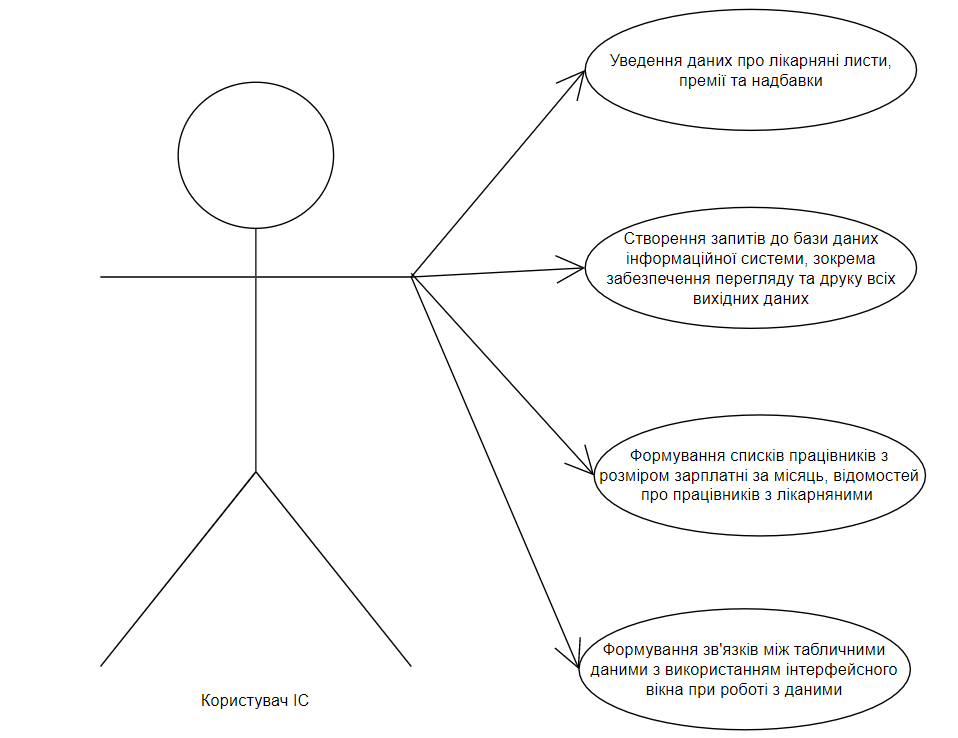


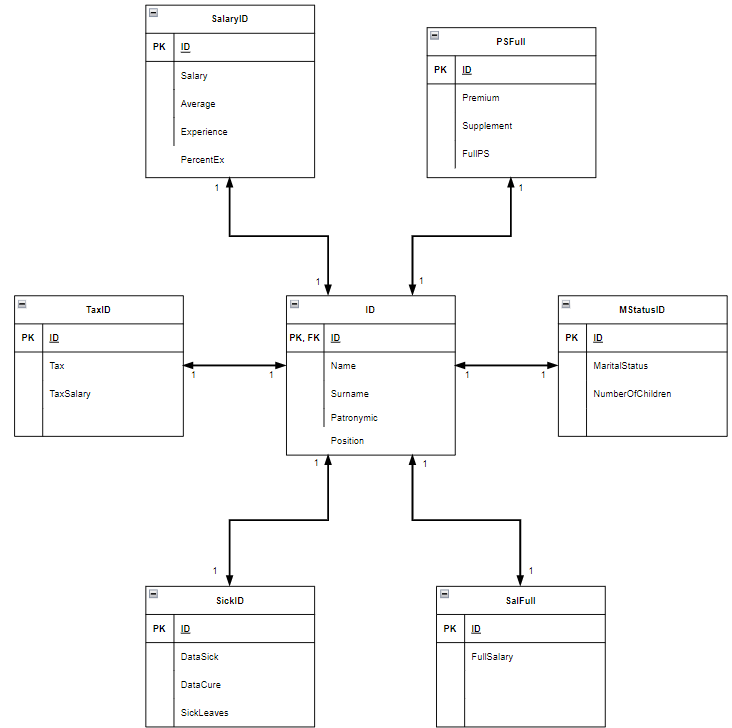
Рис.2.2. Діаграма прецедентів даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства

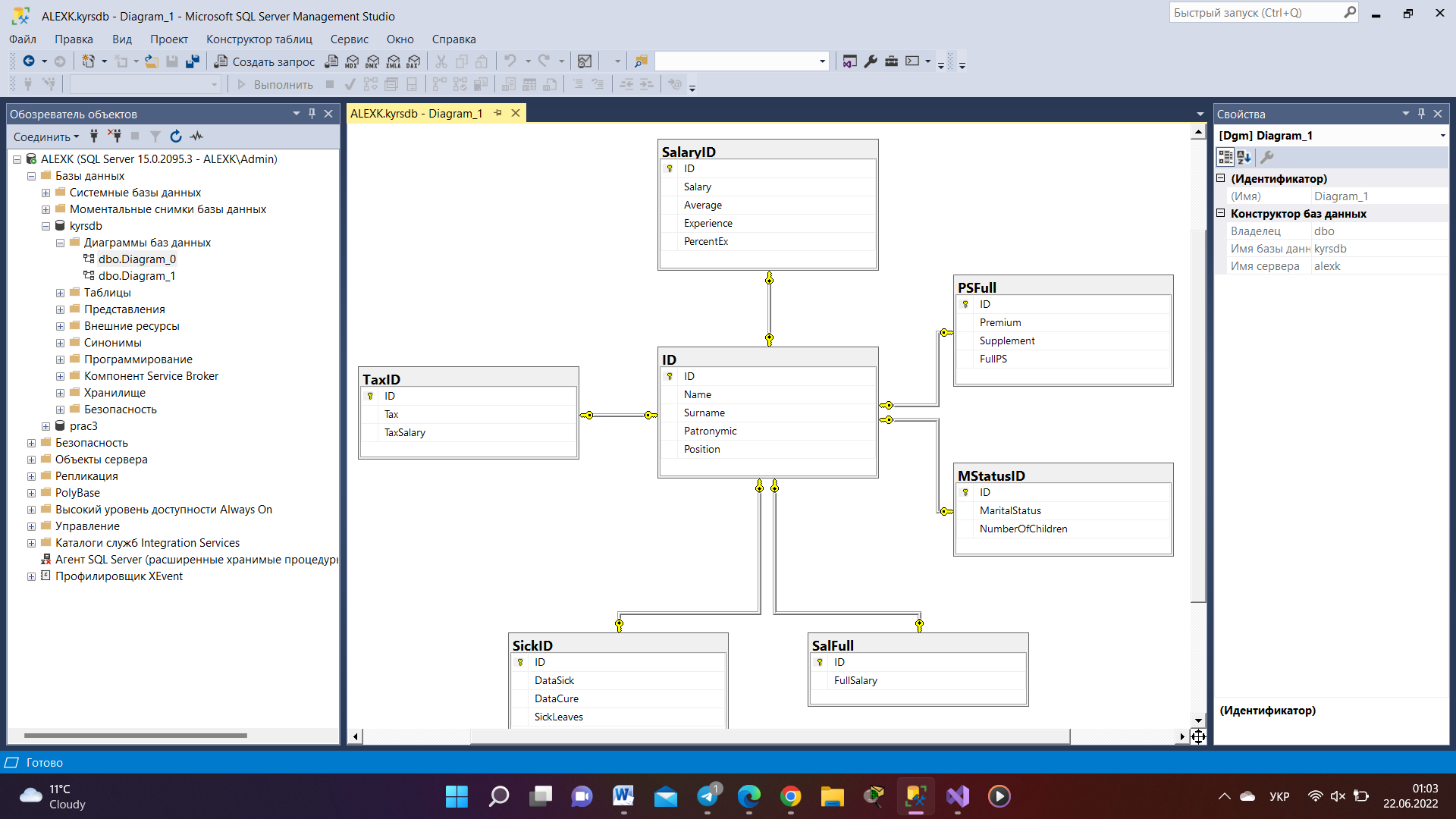
Далі описані концептуальна та фізична моделі проектованої бази даних  інформаційної системи нарахувань заробітної плати .

Виходячи з діаграми прецедентів, потрібно зазначити, що розроблювана  інформаційна система довідкової служби кінотеатрів повинна працювати під  сучасними операційними системами типу Windows. Розрядність операційних  систем – x32 та x64. Об’єм пам’яті, який використовується БД не повинен  перевищувати 2 Gb.

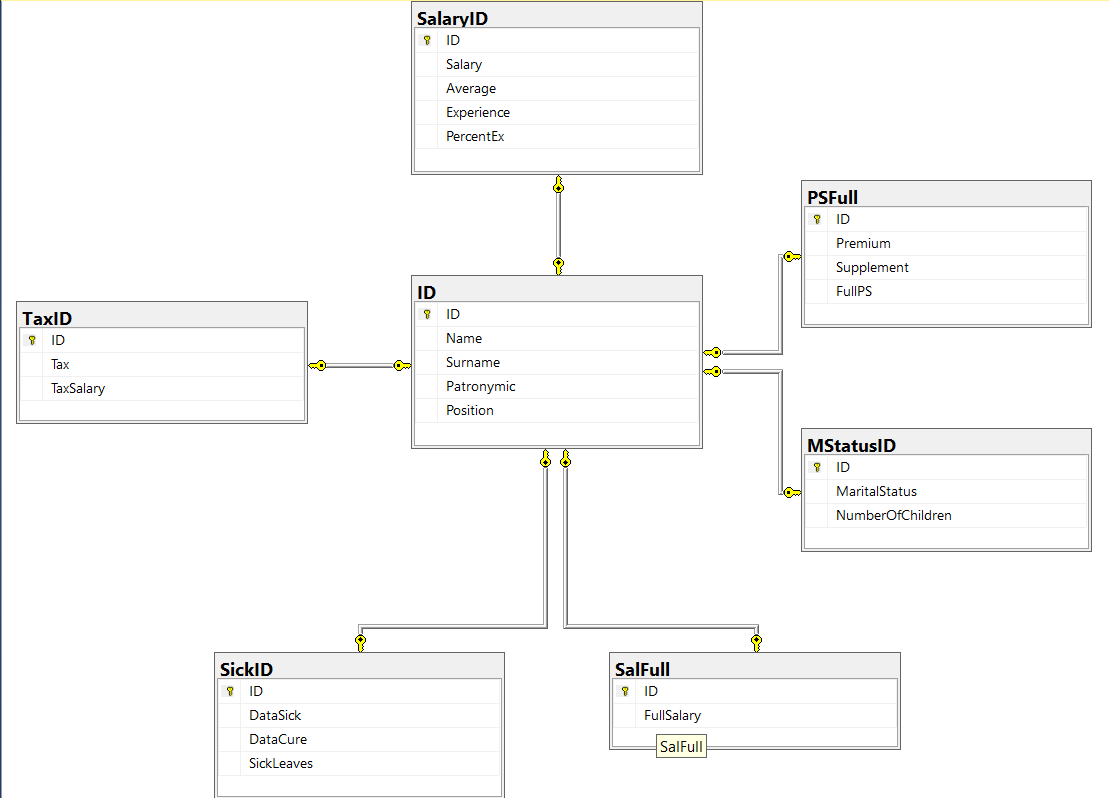
1. **Концептуальна модель бази даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства**

*Концептуальна модель предметної області* – це знання про предметну  область у вигляді понять (концептів). Знання можуть бути як у вигляді  неформальних знань, так і виражені формально за допомогою деяких засобів.  Такими засобами можуть виступати текстові описи предметної області,  набори посадових інструкцій тощо.

На рис. 3.1 показано концептуальну модель бази даних, що розробляється  у даній курсовій роботі.



На рис. 3.2 показана структура бази даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства в середовищі Microsoft SQL Server  Management Studio 18.



На рис. 3.3 наведена  логічна модель бази даниx системи нарахувань заробітної плати в  бухгалтерії підприємства.

1. **Фізична модель бази даних системи нарахувань заробітної плати в  бухгалтерії підприємства**

*Фізична модель* – логічна модель бази даних, виражена в термінах мови  опису даних конкретної СКБД. Обмеження, наявні в логічній моделі даних,  реалізуються різними засобами СКБД, наприклад, за допомогою індексів,  декларативних обмежень цілісності, тригерів, процедур, що зберігаються. При  цьому рішення, прийняті на рівні логічного моделювання, визначають деякі  обмеження, у межах яких можна розвивати фізичну модель даних.

*Створення таблиці ID*

CREATE TABLE [dbo].[ID](

[ID] [int] NOT NULL,

[Name] [varchar](50) NOT NULL,

[Surname] [varchar](50) NOT NULL,

[Patronymic] [varchar](50) NULL,

[Position] [varchar](50) NOT NULL,

CONSTRAINT [PK\_ID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблиці MSatusID*

CREATE TABLE [dbo].[MStatusID](

[ID] [int] NOT NULL,

[MaritalStatus] [varchar](50) NOT NULL,

[NumberOfChildren] [varchar](50) NOT NULL,

CONSTRAINT [PK\_MStatusID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблиці PSFull*

CREATE TABLE [dbo].[PSFull](

[ID] [int] NOT NULL,

[Premium] [float] NOT NULL,

[Supplement] [int] NOT NULL,

[FullPS] [int] NOT NULL,

CONSTRAINT [PK\_PSFull] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблиці SalaryID*

CREATE TABLE [dbo].[SalaryID](

[ID] [int] NOT NULL,

[Salary] [int] NOT NULL,

[Average] [float] NOT NULL,

[Experience] [int] NOT NULL,

[PercentEx] [float] NOT NULL,

CONSTRAINT [PK\_SalaryID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблиці SalFull*

CREATE TABLE [dbo].[SalFull](

[ID] [int] NOT NULL,

[FullSalary] [int] NOT NULL,

CONSTRAINT [PK\_SalFull] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблиці SickID*

CREATE TABLE [dbo].[SickID](

[ID] [int] NOT NULL,

[DataSick] [int] NOT NULL,

[DataCure] [int] NOT NULL,

[SickLeaves] [int] NOT NULL,

CONSTRAINT [PK\_SickID] PRIMARY KEY CLUSTERED

(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

*Створення таблицi TaxID*

CREATE TABLE [dbo].[TaxID](

[ID] [int] NOT NULL,

[Tax] [int] NULL,

[TaxSalary] [int] NULL,

CONSTRAINT [PK\_TaxID] PRIMARY KEY CLUSTERED

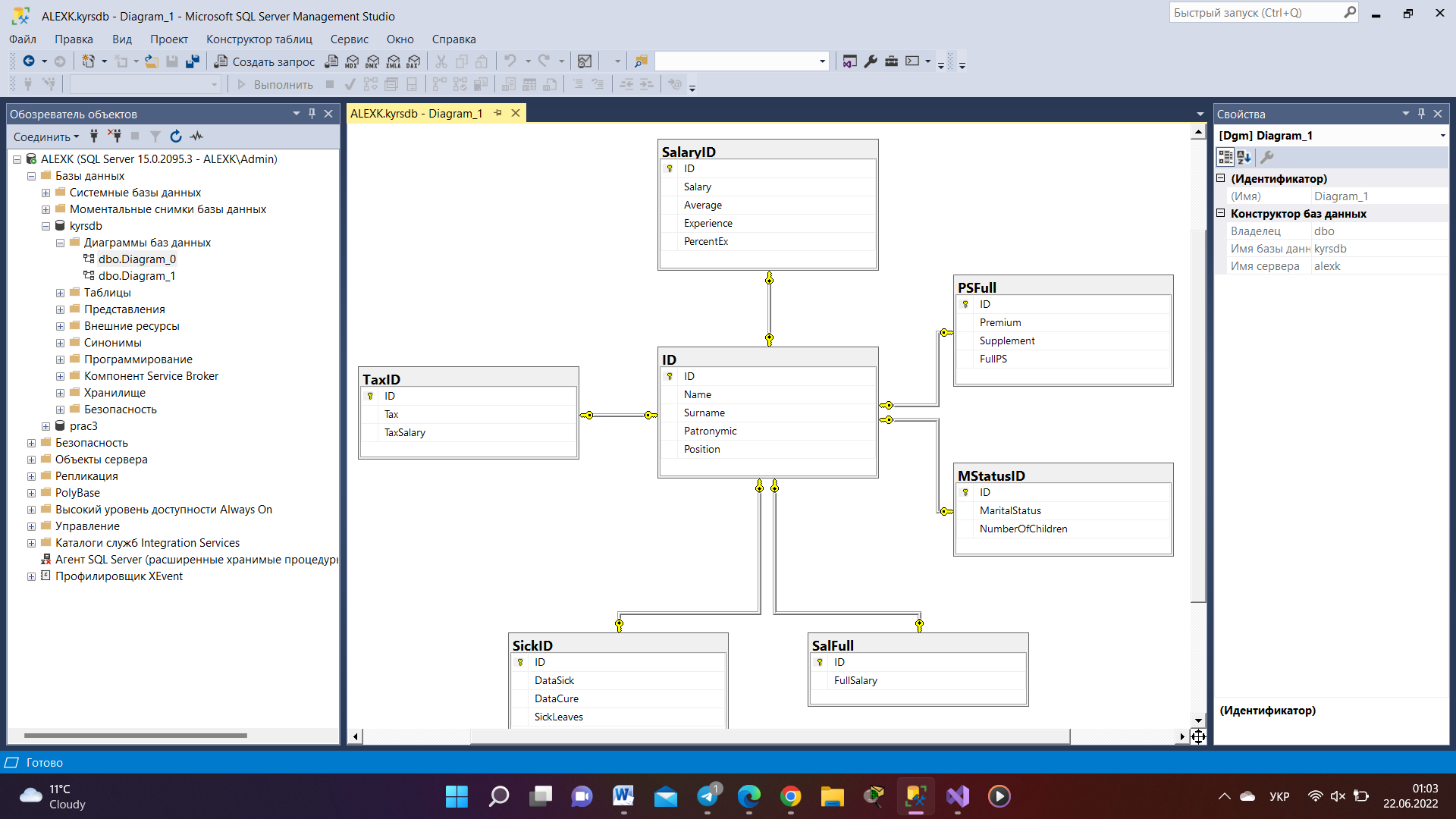
(

[ID] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON, OPTIMIZE\_FOR\_SEQUENTIAL\_KEY = OFF) ON [PRIMARY]

) ON [PRIMARY]

GO

Рис. 4.1. Структура бази даних системи нарахувань заробітної плати в  бухгалтерії підприємства в середовищі Microsoft SQL Server Management Studio 18

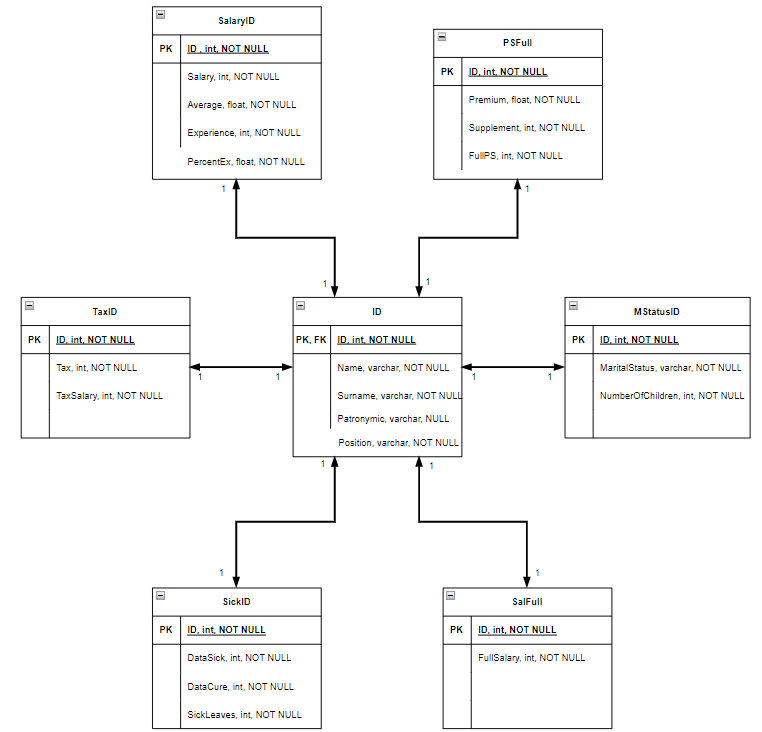
****

Рис. 4.2. Фізична модель бази даних системи нарахувань заробітної плати в  бухгалтерії підприємства

Створена фізична модель показує зв’язки між таблицями вже  спроектованої бази даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства.

1. **Опис реалізації завдань відповідно до обраного варіанту. Створення та структура таблиць бази даних інформаційної  системи нарахувань заробітної плати в бухгалтерії підприємства**

Структури таблиць розробленої бази даних інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства показані у табл. 5.1–5.7.

Таблиця 5.1

Структура таблиці “TaxID”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **Tax** | **int** | **NO** |
| **TaxSalary** | **int** | **NO** |

Таблиця 5.2

Структура таблиці “SalFull”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **FullSalary** | **int** | **NO** |

Таблиця 5.3

Структура таблиці “ID”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK, FK)** | **int** | **NO** |
| **Name** | **varchar(50)** | **NO** |
| **Surname** | **varchar(50)** | **NO** |
| **Patronymic** | **smallint** | **YES** |
| **Position** | **varchar(50)** | **NO** |

Таблиця 5.4

Структура таблиці “MStatusID”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **MaritalStatus** | **varchar(50)** | **NO** |
| **NumberOfChildren** | **varchar(50)** | **NO** |

Таблиця 5.5

Структура таблиці “SalaryID”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **Salary** | **int** | **NO** |
| **Average** | **float** | **NO** |
| **Experience** | **int** | **NO** |
| **PercentEx** | **float** | **NO** |

Таблиця 5.6

Структура таблиці “SickID”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **DataSick** | **int** | **NO** |
| **DataCure** | **int** | **NO** |
| **SickLeaves** | **int** | **NO** |

Таблиця 5.7

Структура таблиці “PSFull”

|  |  |  |
| --- | --- | --- |
| Назва стовпця | Тип | Чи може бути пустим |
| **ID (PK)** | **int** | **NO** |
| **Premium** | **float** | **NO** |
| **Supplement** | **int** | **NO** |
| **FullPS** | **int** | **NO** |

Створені таблиці повністю відображають структуру бази даних, а також  повністю ураховують весь необхідний функціонал при роботі з самою базою  даних інформаційної системи.

## 5.1 Формування запитів до створеної бази даних системи нарахувань заробітної плати в бухгалтерії підприємства

## Нижче наведено основні звіти, які було створено для демонстрації роботи  бази даних інформаційної системи нарахувань заробітної плати. Результати  роботи звітів наведено у додатку Б до даної курсової роботи з урахування  використання інтерфейсу користувача, за допомогою якого деякі запити  керуються параметрами, які вводить користувач бази даних розробленої  інформаційної системи нарахувань заробітної плати в  бухгалтерії підприємства.

Запит 1. Відомості про нарахування кінцевої заробітної плати

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SalaryID.Salary AS Оклад,

dbo.TaxID.TaxSalary AS [Податок пряцівника],

dbo.SickID.SickLeaves AS Лікарняні,

dbo.PSFull.FullPS AS Надбавки,

dbo.SalFull.FullSalary AS Всього

FROM dbo.ID INNER JOIN

dbo.SalFull ON dbo.ID.ID = dbo.SalFull.ID INNER JOIN

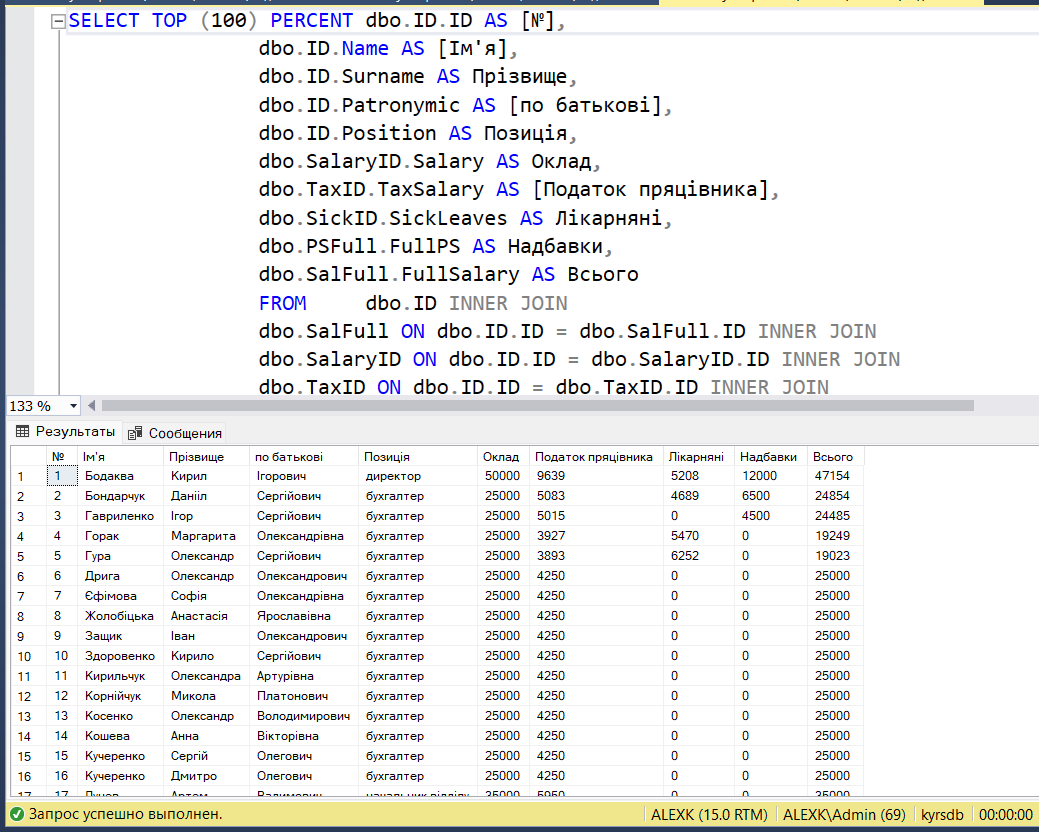
dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID INNER JOIN

dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID INNER JOIN

dbo.SickID ON dbo.ID.ID = dbo.SickID.ID INNER JOIN

dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID

ORDER BY [№]

****

Запит 2. Відомості про середню заробітню плату

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SalaryID.Salary AS Оклад,

dbo.SalaryID.Average AS [У середньому за день],

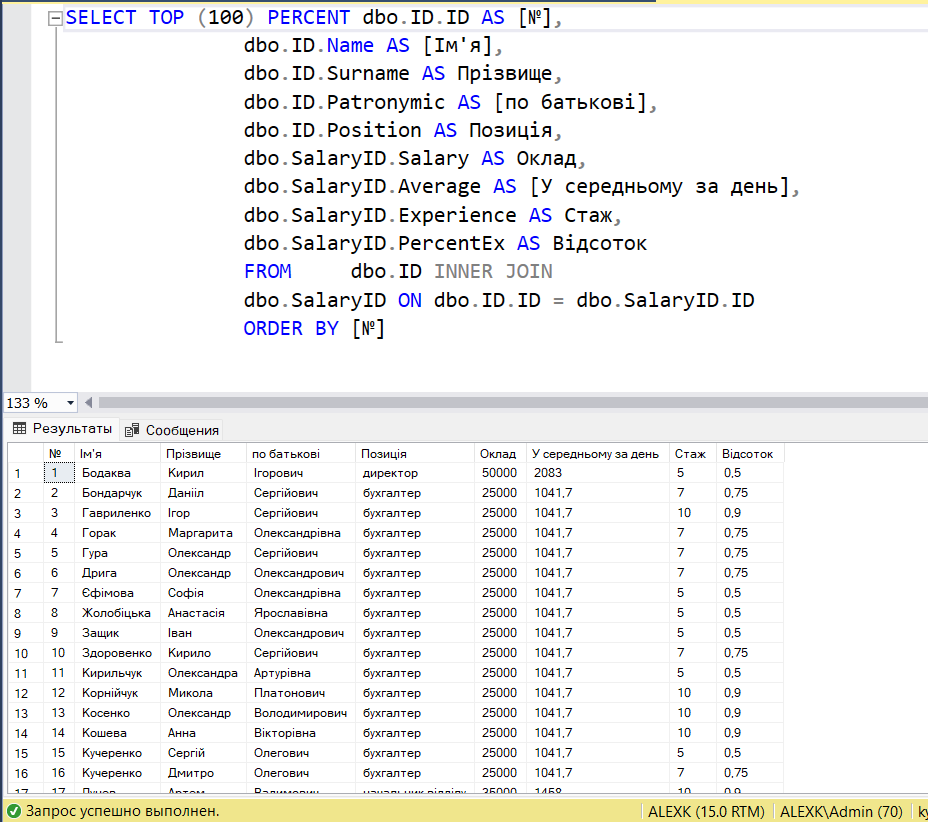
dbo.SalaryID.Experience AS Стаж,

dbo.SalaryID.PercentEx AS Відсоток

FROM dbo.ID INNER JOIN

dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID

ORDER BY [№]

****

Запит 3. Відомості про податки працівників і підприємства.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

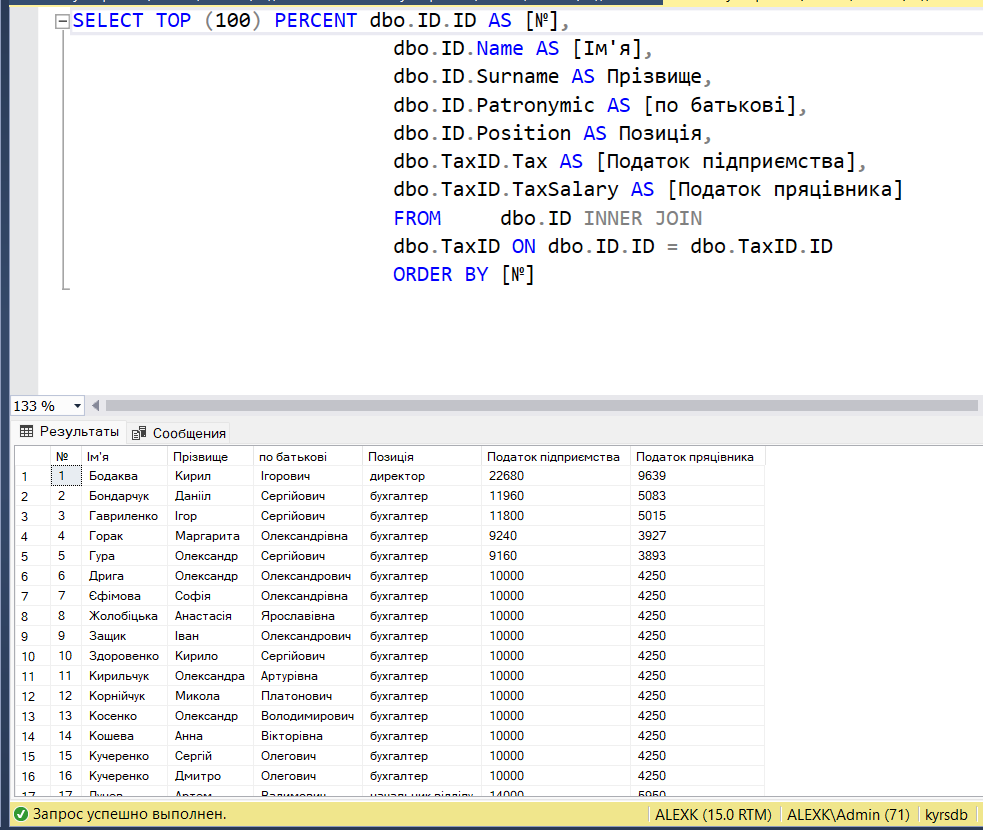
dbo.TaxID.Tax AS [Податок підприємства],

dbo.TaxID.TaxSalary AS [Податок пряцівника]

FROM dbo.ID INNER JOIN

dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID

ORDER BY [№]

****

Запит 4. Відомості про сімейний статус.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

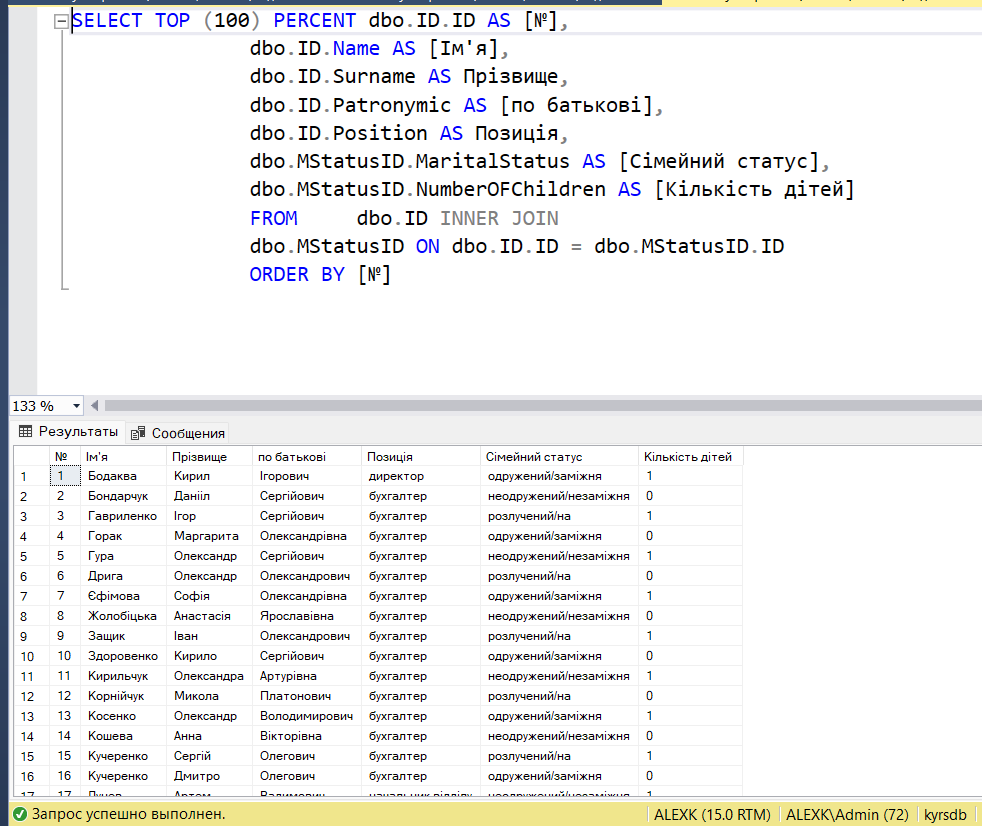
dbo.MStatusID.MaritalStatus AS [Сімейний статус],

dbo.MStatusID.NumberOFChildren AS [Кількість дітей]

FROM dbo.ID INNER JOIN

dbo.MStatusID ON dbo.ID.ID = dbo.MStatusID.ID

ORDER BY [№]

****

Запит 5. Відомості про лікарняні.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SickID.DataSick AS [Дата захворювання],

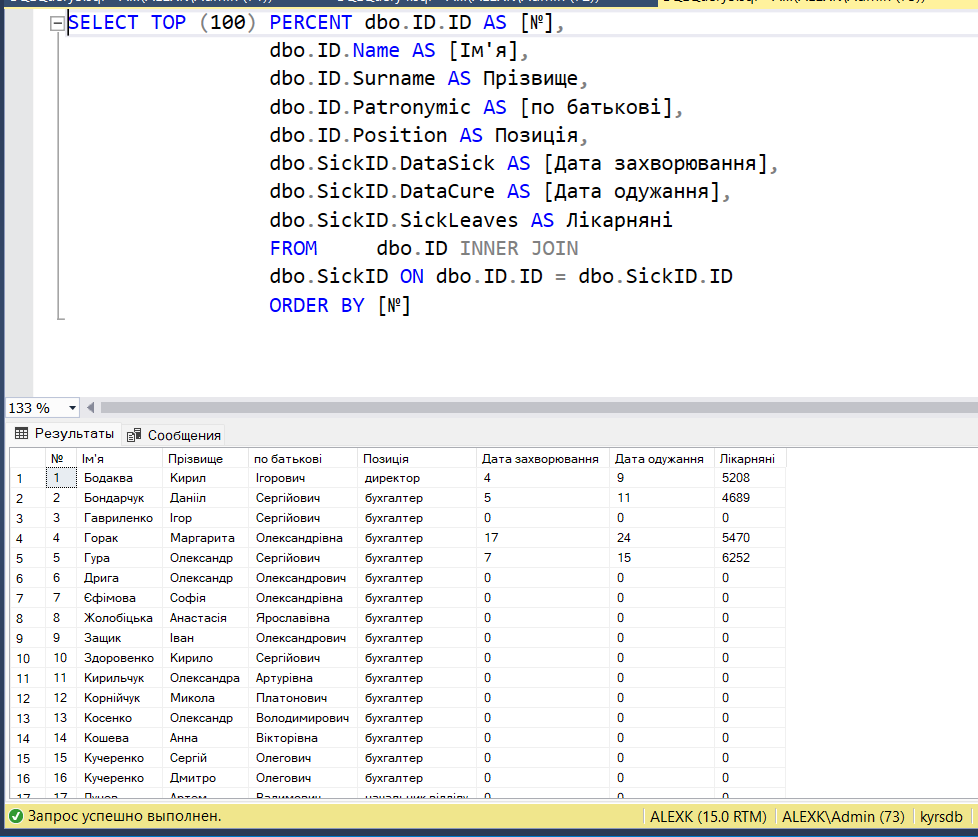
dbo.SickID.DataCure AS [Дата одужання],

dbo.SickID.SickLeaves AS Лікарняні

FROM dbo.ID INNER JOIN

dbo.SickID ON dbo.ID.ID = dbo.SickID.ID

ORDER BY [№]



Запит 6. Відомості про надбавки та премії.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.PSFull.Premium AS Премія,

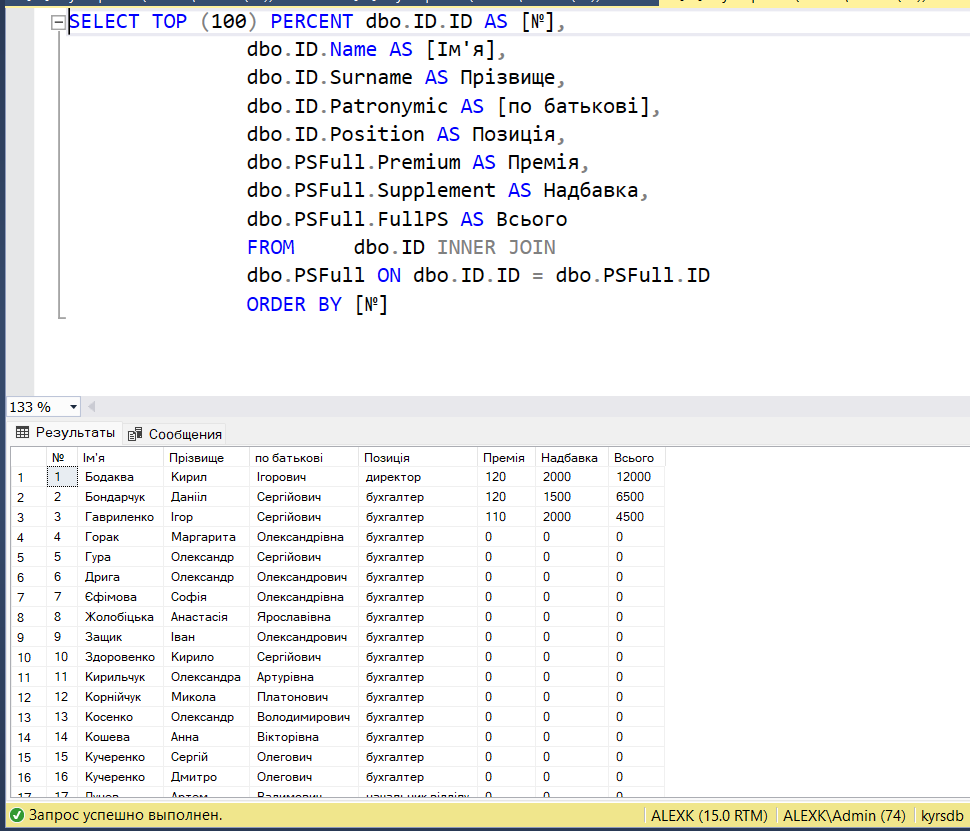
dbo.PSFull.Supplement AS Надбавка,

dbo.PSFull.FullPS AS Всього

FROM dbo.ID INNER JOIN

dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID

ORDER BY [№]

****

Усі розроблені запити демонструють коректну роботу бази даних та

інформаційної системи в цілому.

* 1. **Розроблення інтерфейсної частини для роботи зі створеною  базою даних системи нарахувань заробітної плати в бухгалтерії підприємства**

Інтерфейсна частина інформаційної системи для створеної бази даних  складається з кількох вікон, з якими працює користувач даної інформаційної  системи. З головно вікна користувач має можливість вводити дані про  надбавки, про премії, про дати захворювання та одужання тощо. Кожна форма незалежно  працює з іншими. Усе це забезпечується цілісністю даних, яка забезпечується  відповідними зв’язками у базі, що була спроектована, а також тригерами,  якими вона була доповнена .

Демонстрація роботи тригерів, представлень та процедур наведено у  наступному розділі даної курсової роботи.

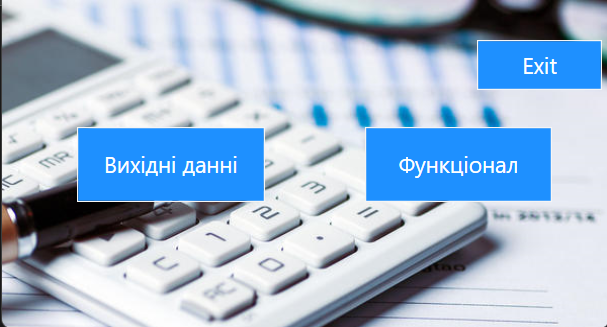


Рис. 5.1. Головне вікно інформаційної системи нарахувань заробітної плати в бухгалтерії підприємства

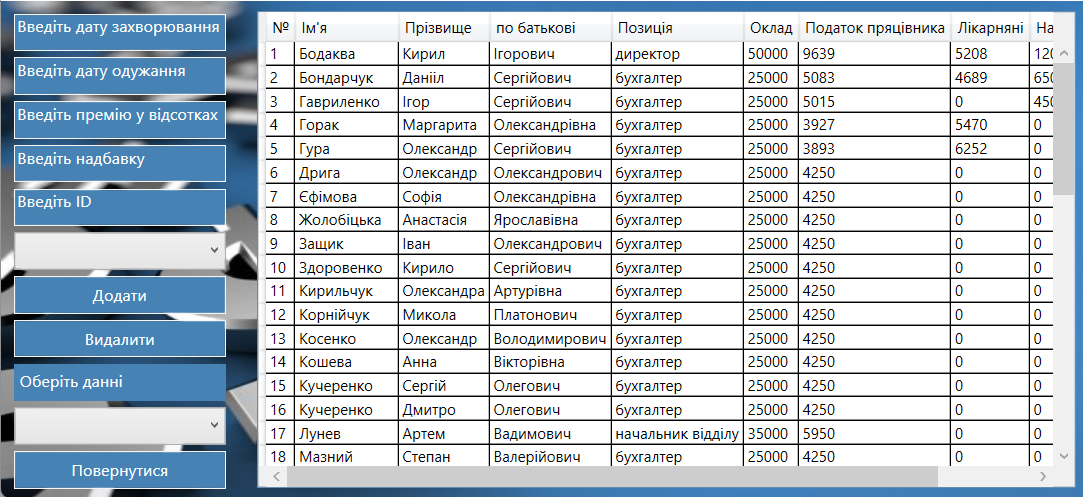
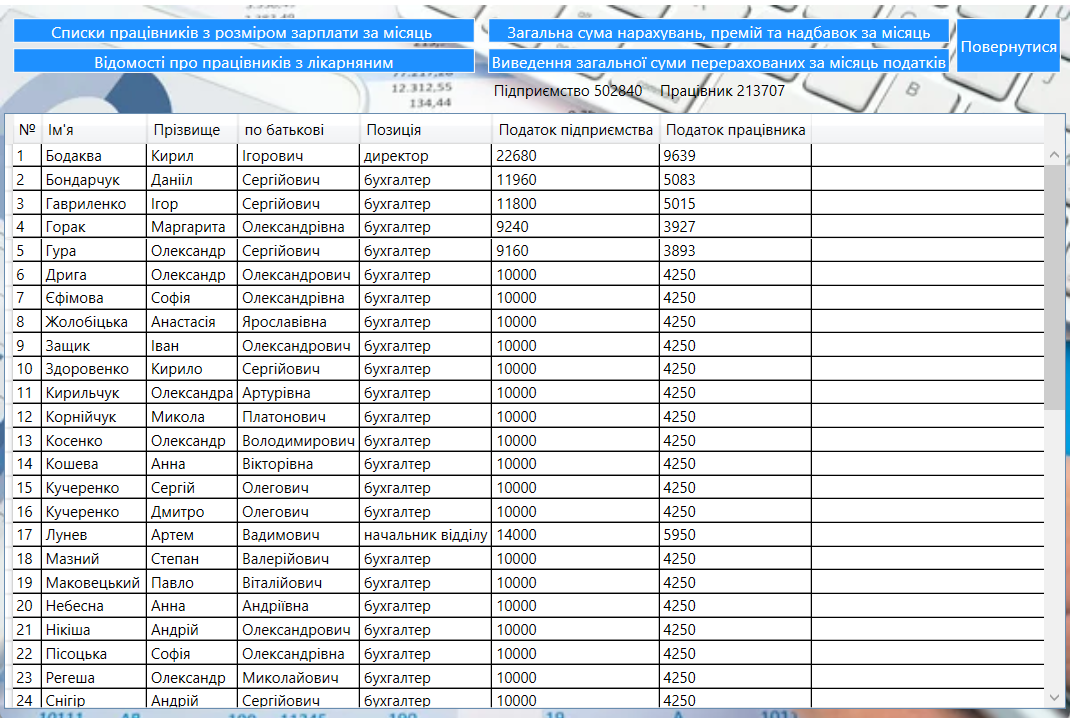


Рис. 5.2. Інтерфейс користувача для редагування даних про премії, надбавки та лікарняні

Рис. 5.3. Інтерфейс користувача для прегляду даних

Створені інтерфейси користувачів дають змогу швидше виконувати  запити до створеної бази даних інформаційної системи, а також дають змогу  відображати дані безпосередньо після їх введення до WPF-форми .

На рисунках 5.1–5.3 показано основні елементи графічного інтерфейсу  користувача за допомогою яких проводиться редагування різних типів даних  з метою забезпечення повного функціонала інформаційної системи.

**6. Тестування розробленої бази даних інформаційної системи нарахувань заробітної плати в бухгалтерії підприємства**

**6.1. Тестування запитів до бази даних**

Нижче наведено основні результати тестування запитів до створеної бази  даних мовою SQL через інтерфейс інформаційної системи.

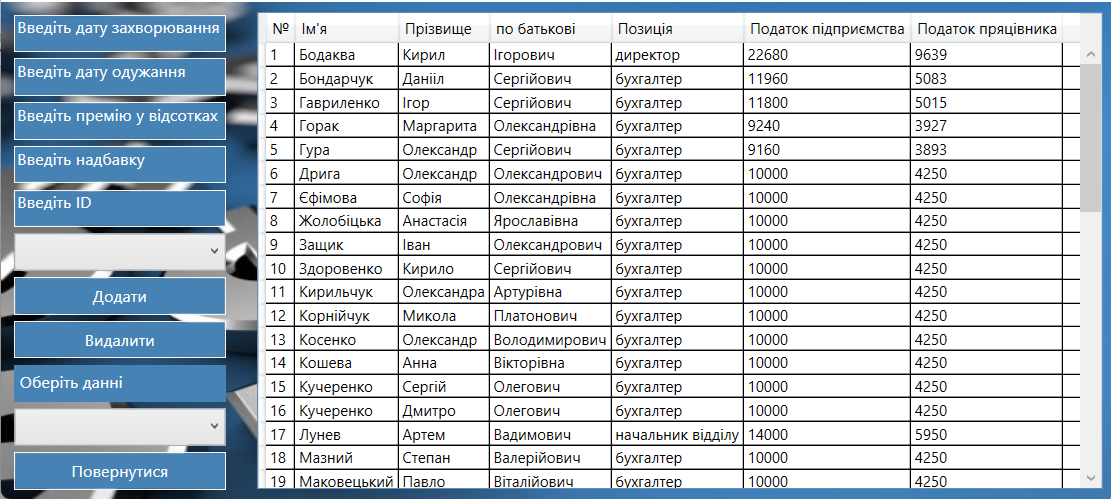
Запит 1



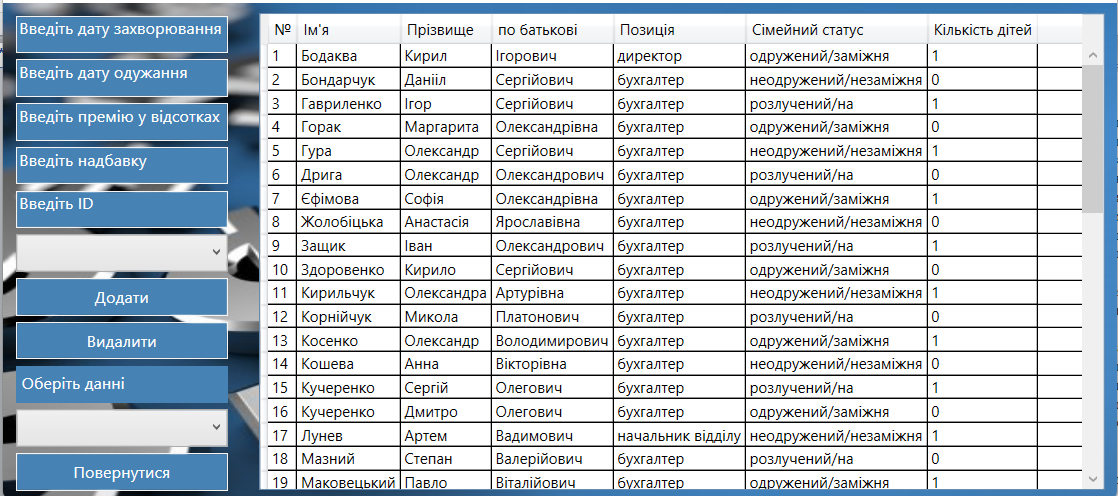
Запит 2



Запит 3

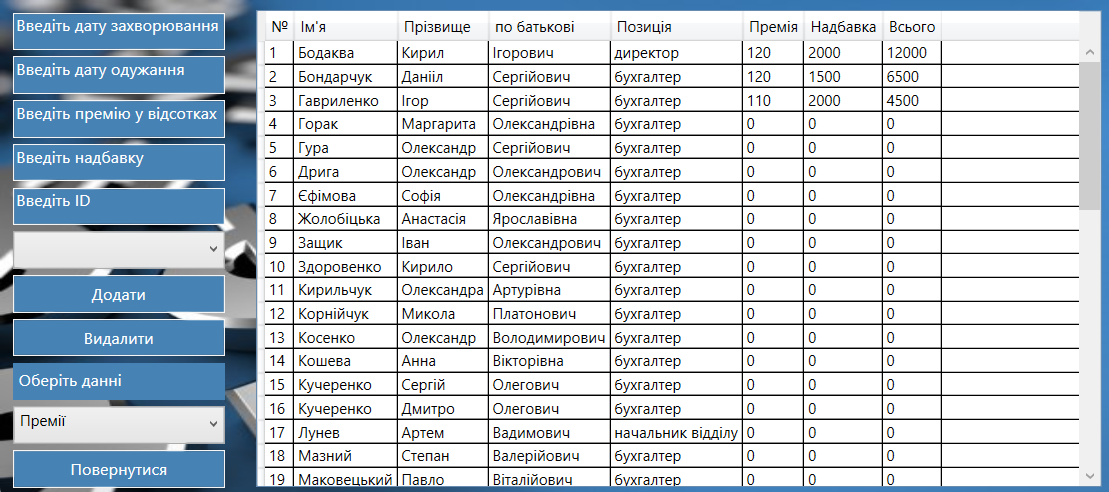


Запит 4



Запит 5 

Запит 6



Як видно з результатів роботи, розроблена база даних інформаційної системи довідкової служби кінотеатрів працює коректно.

**6.2. Тестування представлень та процедур до бази даних**

Для прикладу у роботі реалізовано три представлення.

Представлення показує дані про нарахування заробітної плати працівникам підприємства

CREATE VIEW [dbo].[View\_1]

AS

SELECT dbo.ID.ID, dbo.ID.Name, dbo.ID.Surname, dbo.ID.Position, dbo.PSFull.FullPS, dbo.SalaryID.Salary, dbo.TaxID.TaxSalary, dbo.SickID.SickLeaves, dbo.SalFull.FullSalary

FROM dbo.ID INNER JOIN

dbo.MStatusID ON dbo.ID.ID = dbo.MStatusID.ID INNER JOIN

dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID INNER JOIN

dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID INNER JOIN

dbo.SalFull ON dbo.ID.ID = dbo.SalFull.ID INNER JOIN

dbo.SickID ON dbo.ID.ID = dbo.SickID.ID INNER JOIN

dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID

GO

Результат представлення



|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ID | Name | Surname | Position | FullPS | Salary | TaxSalary | SickLeaves | FullSalary |
|  |  |  |  |  |  |  |  |  |
| 1 | Бодаква | Кирил | директор | 12000 | 50000 | 9639 | 5208 | 47154 |
|  |  |  |  |  |  |  |  |  |
| 2 | Бондарчук | Данііл | бухгалтер | 6500 | 25000 | 5083 | 4689 | 24854 |
|  |  |  |  |  |  |  |  |  |
| 3 | Гавриленко | Ігор | бухгалтер | 4500 | 25000 | 5015 | 0 | 24485 |
|  |  |  |  |  |  |  |  |  |
| 4 | Горак | Маргарита | бухгалтер | 0 | 25000 | 3927 | 5470 | 19249 |
|  |  |  |  |  |  |  |  |  |
| 5 | Гура | Олександр | бухгалтер | 0 | 25000 | 3893 | 6252 | 19023 |
|  |  |  |  |  |  |  |  |  |
| 6 | Дрига | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 7 | Єфімова | Софія | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 8 | Жолобіцька | Анастасія | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 9 | Защик | Іван | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 10 | Здоровенко | Кирило | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 11 | Кирильчук | Олександра | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 12 | Корнійчук | Микола | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 13 | Косенко | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 14 | Кошева | Анна | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 15 | Кучеренко | Сергій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 16 | Кучеренко | Дмитро | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 17 | Лунев | Артем | начальник відділу | 0 | 35000 | 5950 | 0 | 35000 |
|  |  |  |  |  |  |  |  |  |
| 18 | Мазний | Степан | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 19 | Маковецький | Павло | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 20 | Небесна | Анна | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 21 | Нікіша | Андрій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 22 | Пісоцька | Софія | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 23 | Регеша | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 24 | Снігір | Андрій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 25 | Хандохін | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 26 | Яковчук | Артем | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 27 | Асанов | Едем | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 28 | Болюк | Данііл | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 29 | Воробйов | Євгеній | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 30 | Гавриш | Олексій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 31 | Грищенко | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 32 | Дичка | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 33 | Довчунас | Павло | начальник відділу | 0 | 35000 | 5950 | 0 | 35000 |
|  |  |  |  |  |  |  |  |  |
| 34 | Заварзін | Олександр | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 35 | Кулініч | Максим | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 36 | Лавренюк | Андрій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 37 | Мальцев | Микита | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 38 | Махлун | Марія | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 39 | Нечипор | Сергій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 40 | Нижній | Іван | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 41 | Онасенко | Олексій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 42 | Петров | Леонід | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 43 | Пономаренко | Владислава | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 44 | Приймак | Юрій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 45 | Трапезніков | Станіслав | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 46 | Фесюк | Андрій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 47 | Чердаклієв | Кирило | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |
|  |  |  |  |  |  |  |  |  |
| 48 | Якубишин | Анатолій | бухгалтер | 0 | 25000 | 4250 | 0 | 25000 |

*Результати тестування процедури є вірними.*

**6.3. Тестування додавання та виделення даних про працівників за ID**

Інтерфейсна частина виконана у конструкторі..

Функціональна частина, яка підключається до інтерфейсної частини  також реалізована через SQL-запити до створеної бази даних інформаційної  системи нарахувань заробітної плати.

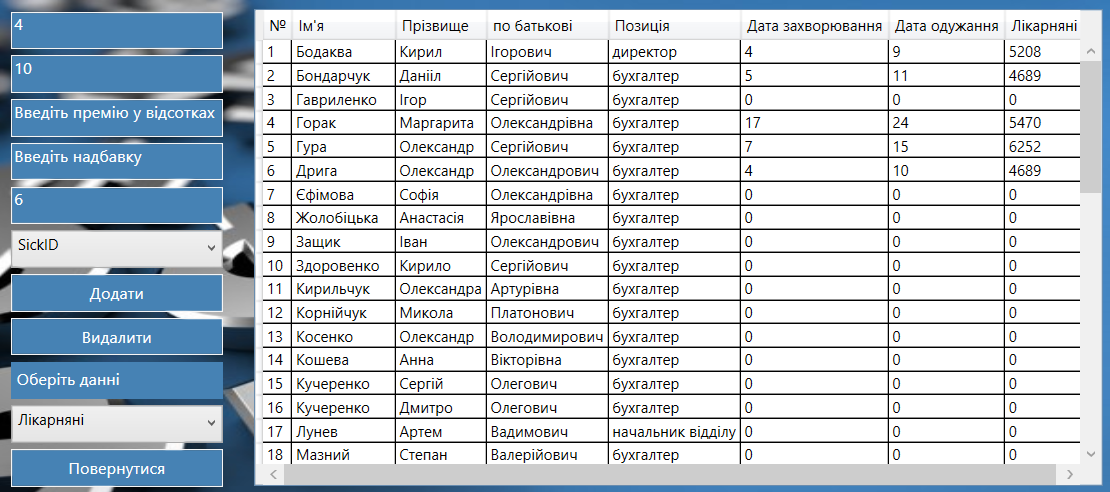


Рис 6.3 Результат додавання дат захворювання і одужання

Рис 6.4 Результат додавання премії та надбавки

У хоті тестування бази даних інформаційної системи помилок її не  виявлено. База даних працює коректно. Інформаційна система має зручний інтерфейс користувача

# *Висновки*

Курсова робота на тему «**Інформаційна система нарахувань заробітної плати в  бухгалтерії підприємства**» може бути застосована практично. Створена програма дозволяє проводити табелювання роботи працівників, нараховувати заробітну платню та додаткові засоби фінансового стимулювання працівників, визначати податкові відрахування та розраховувати суми до сплати працівникам підприємства. Ця програма, як і багато аналогічних програм, може бути застосована практично на будь-якому підприємстві. Але чи здатні подібні програми забезпечити конкурентну перевагу в бізнесі? Ні. Вона вже застаріла безпосередньо на етапі розробки. Станом на сьогодні це радше вхідний квиток до змагань, це must have. Якщо ти не маєш сучасно організованого бухгалтерського відділу , то не варто і розпочинати бізнес. Але я маю на увазі наступне. Саме питання нарахування заробітної платні розпочинається ще на етапі співбесіди з майбутнім працівником. Саме спеціалісти з рекрутингу, або їх ще називають менеджери з персоналу, от саме вони і визначають, яку вартість для компанії має конкретний фахівець та яку суму варто йому заплатити. Тобто, лише в державних компаніях досі можуть залишатись якісь фіксовані ставки виплат за певними посадами. Проте в бізнесі все інакше. Бо програма не може враховувати відмінності психотипів особистості. Одна людина приходить «за грошима», її не цікавить, як називається її посада і таке інше, її цікавить лише цифра «до виплати», і такі співробітники мають порівняно вищу зарплатню. Проте, є діаметрально протилежна категорія працівників, які приходять «за посадою». Така людина може погодитись на значно меншу зарплатню, проте ви повинні надати йому усі можливі елементи статусу, а саме окремий кабінет, велике шкіряне крісло, золоті візитки та табличку «його високість супергіпердиректор» на дверях. Безсумнівно, може статися, що ці дві особистості будуть виконувати майже однакову роботу, проте другий тип для компанії обійдеться дешевше. Також людина може тимчасово погодитись на меншу зарплатню, якщо буде мати перспективу кар’єрного зростання. Потрібно розуміти, що навіть поділ людей на аудиалів, візуалів та кінестетиків має значення. Кінестетику буде приємним сам процес перерахунку грошей руками, тому бажано йому видавати зарплатню невеликими купюрами у віконечко, як це робилось на багатьох підприємствах раніше, щоб він міг насолодитись процесом тримання суми в руках. Тобто, йому можна давати трохи менше, проте дрібними купюрами, але саме давати в руки. Аудиалу бажано дзвонити та сповіщати про суму до отримання. Візуалу присилати електронного листа. Саме визначення психотипу особистості може бути важелем в руках рекрутера для зменшення витрат на персонал. Відповідно, апріорі не може бути застосований сам процес лінійного визначення заробітної платні по посадах. І тут є дуже тонка грань, коли два співробітника дізнаються, що роблять однакову роботу, проте отримують зовсім різні гроші. Є величезна загроза, що в колективі запанує відчуття несправедливості, а це дуже загрозлива річ, бо саме несправедливість спонукає людей радше втратити все, ніж отримувати небагато. А тепер враховуйте, що сам процес перебування персоналу на посадах є динамічним, бо одна людина фахово росте і може вимагати більше відповідальності, проте і більшої оплати. Інша може рухатись до пенсійного віку та не витримувати навантаження. Уявляєте все це в масштабах великого підприємства? Тобто, процес керування персоналом відбувається менеджерами з персоналу значною мірою за рахунок грошової та інших видів винагороди за допомогою фахівців відділу бухгалтерії з використанням сучасного програмного забезпечення. Але чи можна рахувати подібні програми сучасними? Ще раз ні. Ці програми здатні скоротити кількість бухгалтерів. Але сучасна конкуренція, «охота за головами» найкращих фахівців в галузі, динамічне та ефективне керування робітниками є рушійними факторами для значного збільшення кількості менеджерів з персоналу на підприємстві. І все одно процес керування може лише умовно наближатись до оптимального, тому що поза увагою відділу кадрів завжди буде залишатись величезна кількість невідомих факторів, так званих «чорних лебедів».

То чи є можливість скоротити кількість фахівців з персоналу та бухгалтерів, в той же час підтримувати суттєву лояльність своїх провідних фахівців та мінімізувати витрати на їх утримання? Думаю, що так. Проте, потрібно використання значно складніших програм, а в майбутньому, і штучного інтелекту. Складні програми, аналізуючи Big Data, будуть здатні збирати інформацію, визначати психотипи, фінансовий стан, аналізувати минулі місця працевлаштування, тобто, знаходити важелі впливу ще до співбесіди. Рекрутер на співбесіді буде мати чітку картину, хто з кандидатів на посаду може бути максимально корисним за мінімальну ціну і яким чином змусити його погодитись на запропоновану суму грошей. Також, продовжуючи збирати інформацію з Big Data, програма буде безперервно актуалізовувати інформацію про робітника, корегуючи його зарплатню в разі потреби. Повз такої програми не пройде бажання працівника купити нове авто, або взята іпотека, коштовна відпустка та інше. Думаю, що використання подібної програми може бути в чомусь аморальним та містити ознаки маніпулювання, особливо в разі застосування штучного інтелекту. В будь-якому разі, подібне застосування має бути предметом для змістовного обговорення. Але той, хто подібну програму напише, або вже її створив, врешті решт поставить нову планку для всіх без виключення. Таким чином, підсумовуючи, вважаю, що моя програма може розглядатись в найкращому випадку в якості фінішної частини значно потужнішої програми. Або як простий, але примітивний спосіб допомоги бухгалтеру.

**СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ**

1. Kalen, Delaney Inside Microsoft® SQL Server(TM) 2005: Query Tuning and  Optimization. М.: Microsoft Press, 2014. 448 c.

2. Грабер, Мартин SQL для простых смертных. М.: ЛОРИ, 2014. 378 c. 3. Гудсон, Джон Практическое руководство по доступу к данным (+ DVD ROM). М.: БХВ-Петербург, 2013. 304 c.

4. Дунаев, В. В. Базы данных. Язык SQL для студента. М.: БХВ-Петербург,  2016. 288 c.

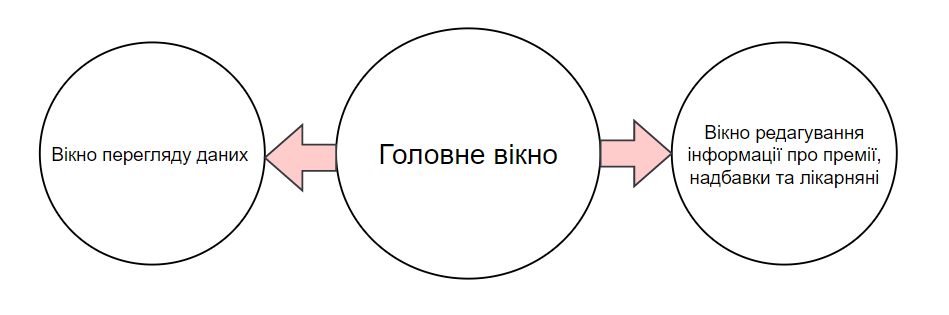
5. Кригель, А. SQL. Библия пользователя. М.: Вильямс, 2013. 110 c. 6. Форта, Бен Освой самостоятельно SQL за 10 минут. М.: Вильямс, 2015.  288 c.

7. Эйри, Джоунс Функции SQL. Справочник программиста. М.: Диалектика  Вильямс, 2014. 556 c.

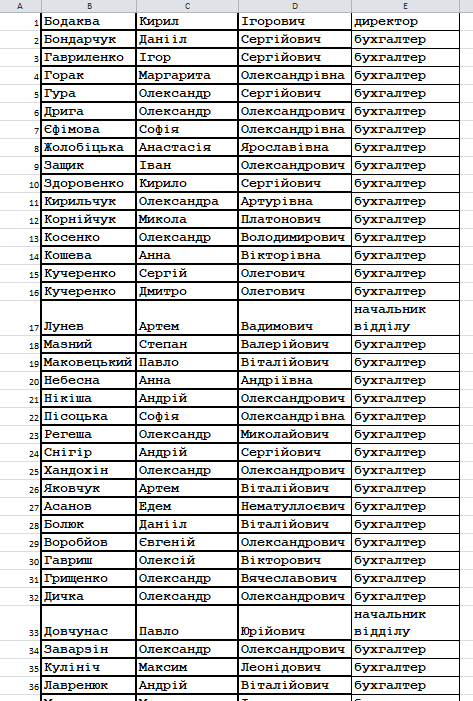
# Додаток А

**Загальні характеристики інформаційної системи**

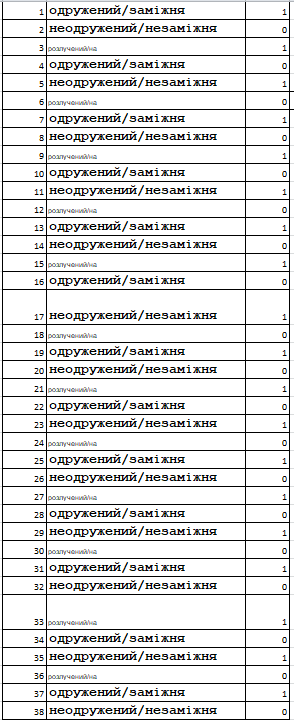
Функціонал інформаційної системи складається з кількох окремих  модулів, загальна залежність між якими показано нижче.

****

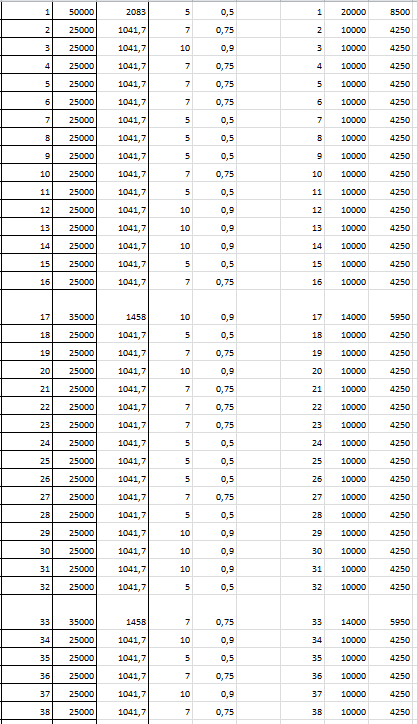
**Рис. А.1**

****

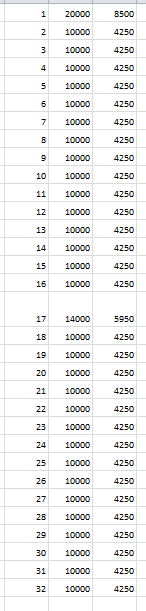
**Рис. А.2 Дані таблиці ID**

****

**Рис. А.3 Дані таблиці MStatusID**

****

**Рис. А.5 Дані таблиці SalaryID**

****

**Рис. А.4 Дані таблиці TaxID**

Усі таблиці, які було показано на рис. А.1–А.5 було реалізовано у MS Excel 2019. Після створення даних таблиць, їх було передано до середовища  MS SQL Server.

Після процедури передавання даних були створені зовнішні ключі для  забезпечення повноцінного функціонування роботи бази даних інформаційної  системи довідкової служби кінотеатрів міста. Окремо було створено тригери  та процедури.

Спроектована та заповнена даними база даних використана WPF додатком, що спроектовано і програмно реалізовано мовою програмування C# у середовищі Visual Studio 2019 Pro.

# Додаток Б

**Запити до розробленої бази даних інформаційної системи**

Запит 1. Відомості про нарахування кінцевої заробітної плати

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SalaryID.Salary AS Оклад,

dbo.TaxID.TaxSalary AS [Податок пряцівника],

dbo.SickID.SickLeaves AS Лікарняні,

dbo.PSFull.FullPS AS Надбавки,

dbo.SalFull.FullSalary AS Всього

FROM dbo.ID INNER JOIN

dbo.SalFull ON dbo.ID.ID = dbo.SalFull.ID INNER JOIN

dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID INNER JOIN

dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID INNER JOIN

dbo.SickID ON dbo.ID.ID = dbo.SickID.ID INNER JOIN

dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID

ORDER BY [№]

Запит 2. Відомості про середню заробітню плату

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SalaryID.Salary AS Оклад,

dbo.SalaryID.Average AS [У середньому за день],

dbo.SalaryID.Experience AS Стаж,

dbo.SalaryID.PercentEx AS Відсоток

FROM dbo.ID INNER JOIN

dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID

ORDER BY [№]

Запит 3. Відомості про податки працівників і підприємства.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.TaxID.Tax AS [Податок підприємства],

dbo.TaxID.TaxSalary AS [Податок пряцівника]

FROM dbo.ID INNER JOIN

dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID

ORDER BY [№]

Запит 4. Відомості про сімейний статус.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.MStatusID.MaritalStatus AS [Сімейний статус],

dbo.MStatusID.NumberOFChildren AS [Кількість дітей]

FROM dbo.ID INNER JOIN

dbo.MStatusID ON dbo.ID.ID = dbo.MStatusID.ID

ORDER BY [№]

Запит 5. Відомості про лікарняні.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.SickID.DataSick AS [Дата захворювання],

dbo.SickID.DataCure AS [Дата одужання],

dbo.SickID.SickLeaves AS Лікарняні

FROM dbo.ID INNER JOIN

dbo.SickID ON dbo.ID.ID = dbo.SickID.ID

ORDER BY [№]

Запит 6. Відомості про надбавки та премії.

SELECT TOP (100) PERCENT dbo.ID.ID AS [№],

dbo.ID.Name AS [Ім'я],

dbo.ID.Surname AS Прізвище,

dbo.ID.Patronymic AS [по батькові],

dbo.ID.Position AS Позиція,

dbo.PSFull.Premium AS Премія,

dbo.PSFull.Supplement AS Надбавка,

dbo.PSFull.FullPS AS Всього

FROM dbo.ID INNER JOIN

dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID

ORDER BY [№]

# Додаток В

**Програмна реалізація інтерфейсної частини інформаційної системи  та її тестування**

*Лістинг В.1. Функціональна програмна реалізація WPF головного вікна  розробленої інформаційної системи.*

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

namespace WpfApp2

{

/// <summary>

/// Логика взаимодействия для MainWindow.xaml

/// </summary>

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

Window1 w1 = new Window1();

Hide();

w1.Show();

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

System.Windows.Application.Current.Shutdown();

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

Window2 w2 = new Window2();

Hide();

w2.Show();

}

}

}

*Лістинг В.2. Інтерфейсна XML-реалізація WPF головного вікна  розробленої інформаційної системи.*

<Window x:Class="WpfApp2.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="" Height="300" Width="500" Background="DarkSeaGreen">

<Grid>

<Grid.Background>

<ImageBrush ImageSource="/img2.jpg"/>

</Grid.Background>

<Button Content="Функціонал" HorizontalAlignment="Left" Margin="290,0,0,0" VerticalAlignment="Center" Height="60" Width="150" Click="Button\_Click\_2" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="18" Foreground="White"/>

<Button Content="Exit" HorizontalAlignment="Left" Margin="380,32,0,0" VerticalAlignment="Top" Height="40" Width="100" Click="Button\_Click\_1" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="18" Foreground="White"/>

<Button Content="Вихідні данні" HorizontalAlignment="Left" Margin="60,0,0,0" VerticalAlignment="Center" Height="60" Width="150" Click="Button\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="18" Foreground="White"/>

</Grid>

</Window>

*Лістинг В.3. Функціональна програмна реалізація WPF вихідного вікна розробленої інформаційної системи.*

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp2

{

/// <summary>

/// Логика взаимодействия для Window1.xaml

/// </summary>

public partial class Window1 : Window

{

string connectionString = null;

SqlConnection connection = null;

SqlCommand command;

SqlDataAdapter adapter;

//Data Source=AlexK;Initial Catalog=kyrsdb;Integrated Security=True

public Window1()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

}

private void GetAndShowData(string SQLQuery, DataGrid dataGrid)

{

connection = new SqlConnection(connectionString);

connection.Open();

command = new SqlCommand(SQLQuery, connection);

adapter = new SqlDataAdapter(command);

DataTable Table = new DataTable();

adapter.Fill(Table);

dataGrid.ItemsSource = Table.DefaultView;

connection.Close();

}

public object RunSQL(string sql)

{

connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

return command.ExecuteScalar();

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

SumTax.Content = null;

SumTaxSal.Content = null;

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SalFull.FullSalary AS Зарплата " +

"FROM dbo.ID INNER JOIN " +

"dbo.SalFull ON dbo.ID.ID = dbo.SalFull.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, DataGrid1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_1(object sender, RoutedEventArgs e)

{

SumTax.Content = null;

SumTaxSal.Content = null;

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SickID.DataSick AS [Дати захворювання], " +

"dbo.SickID.DataCure AS [Дати одужання] " +

"FROM dbo.ID INNER JOIN " +

"dbo.SickID ON dbo.ID.ID = dbo.SickID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, DataGrid1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_2(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№]," +

"dbo.ID.Name AS [Ім'я]," +

"dbo.ID.Surname AS Прізвище," +

"dbo.ID.Patronymic AS [по батькові]," +

"dbo.ID.Position AS Позиція," +

"dbo.TaxID.Tax AS [Податок підприємства], " +

"dbo.TaxID.TaxSalary AS [Податок працівника] " +

"FROM dbo.ID INNER JOIN " +

"dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID " +

"ORDER BY [№]";

try

{

GetAndShowData(sqlQ, DataGrid1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

SumTax.Content = "Підприємство " + Convert.ToString(RunSQL("SELECT SUM(Tax) FROM TaxID"));

SumTaxSal.Content = "Працівник " + Convert.ToString(RunSQL("SELECT SUM(TaxSalary) FROM TaxID"));

}

private void Button\_Click\_3(object sender, RoutedEventArgs e)

{

SumTax.Content = null;

SumTaxSal.Content = null;

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція," +

"dbo.PSFull.FullPS AS [Сума нарахувань] " +

"FROM dbo.ID INNER JOIN " +

"dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID " +

"ORDER BY [№]";

try

{

GetAndShowData(sqlQ, DataGrid1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Button\_Click\_4(object sender, RoutedEventArgs e)

{

MainWindow mw = new MainWindow();

Hide();

mw.Show();

}

}

}

*Лістинг В.4. Інтерфейсна XML-реалізація WPF вихідного вікна  розробленої інформаційної системи.*

<Window x:Class="WpfApp2.Window1"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="" Height="610" Width="870">

<Grid Margin="0,0,0,0">

<Grid.Background>

<ImageBrush ImageSource="/img3.jpg"/>

</Grid.Background>

<Button Content="Списки працівників з розміром зарплати за місяць " HorizontalAlignment="Left" Margin="10,10,0,0" VerticalAlignment="Top" Width="370" Height="20" Click="Button\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Bідомості про працівників з лікарняним" HorizontalAlignment="Left" Margin="10,34,0,0" VerticalAlignment="Top" Width="370" Height="20" Click="Button\_Click\_1" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Загальна сума нарахувань, премій та надбавок за місяць" HorizontalAlignment="Left" Margin="390,10,0,0" VerticalAlignment="Top" Width="370" Height="20" Click="Button\_Click\_3" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Виведення загальної суми перерахованих за місяць податків" HorizontalAlignment="Left" Margin="390,34,0,0" VerticalAlignment="Top" Width="370" Height="20" Click="Button\_Click\_2" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<DataGrid x:Name="DataGrid1" HorizontalAlignment="Center" Margin="0,86,0,10" Width="850" />

<Button Content="Повернутися" HorizontalAlignment="Left" Margin="765,10,0,0" VerticalAlignment="Top" Height="44" Width="84" Click="Button\_Click\_4" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Label x:Name="SumTax" Content="" HorizontalAlignment="Left" Margin="390,55,0,0" VerticalAlignment="Top" RenderTransformOrigin="1.069,1.311"/>

<Label x:Name="SumTaxSal" Content="" HorizontalAlignment="Left" Margin="523,55,0,0" VerticalAlignment="Top" RenderTransformOrigin="0.049,0.079"/>

</Grid>

</Window>

*Лістинг В.1. Функціональна програмна реалізація WPF фунціонального вікна  розробленої інформаційної системи.*

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Shapes;

namespace WpfApp2

{

/// <summary>

/// Логика взаимодействия для Window2.xaml

/// </summary>

public partial class Window2 : Window

{

string connectionString = null;

SqlConnection connection = null;

SqlCommand command;

SqlDataAdapter adapter;

//Data Source=AlexK;Initial Catalog=kyrsdb;Integrated Security=True

DataTable dT1;

String strQ;

public Window2()

{

InitializeComponent();

connectionString = ConfigurationManager.ConnectionStrings["DefaultConnection"].ConnectionString;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

connection = new SqlConnection(connectionString);

connection.Open();

if (connection.State == System.Data.ConnectionState.Open)

{

try

{

string from = DelFrom.Text;

if (from == "SickID")

{

String Id = IDField.Text;

strQ = "";

strQ += "UPDATE SickID SET DataCure = '" + 0 + "' , DataSick = '" + 0 + "' , SickLeaves = '" + 0 + "' WHERE ID = '" + Id + "';";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

connection.Close();

strQ = null;

FullSalarY(Convert.ToInt32(Id));

}

else if (from == "SalaryID")

{

String Id = IDField.Text;

strQ = "";

strQ += "UPDATE PSFull SET Premium = '" + 0 + "' , Supplement = '" + 0 + "' , FullPS = '" + 0 + "' WHERE ID = '" + Id + "';";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

connection.Close();

strQ = null;

FullSalarY(Convert.ToInt32(Id));

}

}

catch { }

}

}

public object RunSQL(string sql)

{

connection = new SqlConnection(connectionString);

connection.Open();

SqlCommand command = new SqlCommand(sql, connection);

return command.ExecuteScalar();

}

private void Add\_Click(object sender, RoutedEventArgs e)

{

connection = new SqlConnection(connectionString);

connection.Open();

if (connection.State == System.Data.ConnectionState.Open)

{

string from = DelFrom.Text;

try

{

if (from == "SickID")

{

String Id, DataS, DataC;

Id = IDField.Text;

DataS = DSick.Text;

DataC = Dcure.Text;

int Exp = Convert.ToInt32(RunSQL("SELECT Experience from SalaryID where ID = " + Id));

double Per = Convert.ToDouble(RunSQL("SELECT PercentEx from SalaryID where ID = " + Id));

int Aver = Convert.ToInt32(RunSQL("SELECT Average from SalaryID where ID = " + Id));

int numofd = Convert.ToInt32(DataC) - Convert.ToInt32(DataS);

int sickl = Convert.ToInt32(1.0 \* numofd \* Aver \* Per);

strQ = "";

strQ += "UPDATE SickID SET DataCure = " + DataC + " , DataSick = " + DataS + " , SickLeaves = " + sickl.ToString() + " WHERE ID = '" + Id + "';";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

connection.Close();

strQ = null;

FullSalarY(Convert.ToInt32(Id));

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SickID.DataSick AS [Дата захворювання], " +

"dbo.SickID.DataCure AS [Дата одужання], " +

"dbo.SickID.SickLeaves AS Лікарняні " +

"FROM dbo.ID INNER JOIN " +

"dbo.SickID ON dbo.ID.ID = dbo.SickID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

IDField.Text = "Введіть ID";

DSick.Text = "Введіть дати захворювання";

Dcure.Text = "Введіть дату одужання";

}

else if (from == "SalaryID")

{

String Id, Pre, Sup;

Id = IDField.Text;

Pre = Prem.Text;

Sup = Nadb.Text;

int salary = Convert.ToInt32(RunSQL("SELECT Salary from SalaryID where ID = " + Id));

int fpr = Convert.ToInt32((salary \* Convert.ToInt32(Pre) / 100 + Convert.ToInt32(Sup)) - salary);

strQ = "";

strQ += "UPDATE PSFull SET Premium = " + Pre + " , Supplement = " + Sup + " , FullPS = " + fpr.ToString() + " WHERE ID = '" + Id + "';";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

connection.Close();

strQ = null;

FullSalarY(Convert.ToInt32(Id));

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.PSFull.Premium AS Премія, " +

"dbo.PSFull.Supplement AS Надбавка, " +

"dbo.PSFull.FullPS AS Всього " +

"FROM dbo.ID INNER JOIN " +

"dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

IDField.Text = "Введіть ID";

Prem.Text = "Введіть премію у відсотках";

Nadb.Text = "Введіть надбавку";

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

public void FullSalarY(int Id)

{

connection = new SqlConnection(connectionString);

connection.Open();

if (connection.State == System.Data.ConnectionState.Open)

{

int fpr = Convert.ToInt32(RunSQL("SELECT FullPS from PSFull where ID = " + Id));

int salary = Convert.ToInt32(RunSQL("SELECT Salary from SalaryID where ID = " + Id));

int dataC = Convert.ToInt32(RunSQL("SELECT DataCure from SickID where ID = " + Id));

int dataS = Convert.ToInt32(RunSQL("SELECT DataSick from SickID where ID = " + Id));

int Aver = Convert.ToInt32(RunSQL("SELECT Average from SalaryID where ID = " + Id));

int sickl = Convert.ToInt32(RunSQL("SELECT SickLeaves from SickID where ID = " + Id));

int S = salary - (dataC - dataS) \* Aver + sickl + fpr;

int tax, tax1;

tax1 = S / 100 \* 40;

if (S > 20000)

tax = S / 100 \* 17;

else

tax = S / 100 \* 15;

S -= tax;

strQ = "";

strQ += "UPDATE TaxID SET TaxSalary = " + tax.ToString() + "WHERE ID = '" + Id + "'; ";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

strQ = "";

strQ += "UPDATE TaxID SET Tax = " + tax1.ToString() + "WHERE ID = '" + Id + "'; ";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

strQ = "";

strQ += "UPDATE SalFull SET FullSalary = " + S.ToString() + "WHERE ID = '" + Id + "'; ";

command = new SqlCommand(strQ, connection);

MessageBox.Show(command.ExecuteNonQuery().ToString());

connection.Close();

}

}

private void Return\_Click(object sender, RoutedEventArgs e)

{

MainWindow mw = new MainWindow();

Hide();

mw.Show();

}

private void FullSalary\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SalaryID.Salary AS Оклад, " +

"dbo.TaxID.TaxSalary AS [Податок пряцівника], " +

"dbo.SickID.SickLeaves AS Лікарняні, " +

"dbo.PSFull.FullPS AS Надбавки, " +

"dbo.SalFull.FullSalary AS Всього " +

"FROM dbo.ID INNER JOIN " +

"dbo.SalFull ON dbo.ID.ID = dbo.SalFull.ID INNER JOIN " +

"dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID INNER JOIN " +

"dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID INNER JOIN " +

"dbo.SickID ON dbo.ID.ID = dbo.SickID.ID INNER JOIN " +

"dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Salary\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SalaryID.Salary AS Оклад, " +

"dbo.SalaryID.Average AS [У середньому за день], " +

"dbo.SalaryID.Experience AS Стаж, " +

"dbo.SalaryID.PercentEx AS Відсоток " +

"FROM dbo.ID INNER JOIN " +

"dbo.SalaryID ON dbo.ID.ID = dbo.SalaryID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Tax\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.TaxID.Tax AS [Податок підприємства], " +

"dbo.TaxID.TaxSalary AS [Податок пряцівника] " +

"FROM dbo.ID INNER JOIN " +

"dbo.TaxID ON dbo.ID.ID = dbo.TaxID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void MarSt\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.MStatusID.MaritalStatus AS [Сімейний статус], " +

"dbo.MStatusID.NumberOFChildren AS [Кількість дітей] " +

"FROM dbo.ID INNER JOIN " +

"dbo.MStatusID ON dbo.ID.ID = dbo.MStatusID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Sick\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.SickID.DataSick AS [Дата захворювання], " +

"dbo.SickID.DataCure AS [Дата одужання], " +

"dbo.SickID.SickLeaves AS Лікарняні " +

"FROM dbo.ID INNER JOIN " +

"dbo.SickID ON dbo.ID.ID = dbo.SickID.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void Prem\_Click(object sender, RoutedEventArgs e)

{

string sqlQ = "SELECT TOP (100) PERCENT dbo.ID.ID AS [№], " +

"dbo.ID.Name AS [Ім'я], " +

"dbo.ID.Surname AS Прізвище, " +

"dbo.ID.Patronymic AS [по батькові], " +

"dbo.ID.Position AS Позиція, " +

"dbo.PSFull.Premium AS Премія, " +

"dbo.PSFull.Supplement AS Надбавка, " +

"dbo.PSFull.FullPS AS Всього " +

"FROM dbo.ID INNER JOIN " +

"dbo.PSFull ON dbo.ID.ID = dbo.PSFull.ID " +

"ORDER BY [№] ";

try

{

GetAndShowData(sqlQ, dt1);

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void GetAndShowData(string SQLQuery, DataGrid dataGrid)

{

connection = new SqlConnection(connectionString);

connection.Open();

command = new SqlCommand(SQLQuery, connection);

adapter = new SqlDataAdapter(command);

DataTable Table = new DataTable();

adapter.Fill(Table);

dataGrid.ItemsSource = Table.DefaultView;

connection.Close();

}

}

}

*Лістинг В.2. Інтерфейсна XML-реалізація WPF функціонального вікна  розробленої інформаційної системи.*

<Window x:Class="WpfApp2.Window2"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:WpfApp2"

mc:Ignorable="d"

Title="" Height="435" Width="884">

<Grid Margin="0,0,0,0">

<Grid.Background>

<ImageBrush ImageSource="/img4.jpg"/>

</Grid.Background>

<TextBox x:Name="Prem" HorizontalAlignment="Left" Margin="10,80,0,0" TextWrapping="Wrap" Text="Введіть премію у відсотках" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<TextBox x:Name="DSick" HorizontalAlignment="Left" Margin="10,10,0,0" TextWrapping="Wrap" Text="Введіть дату захворювання" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White" />

<TextBox x:Name="Dcure" HorizontalAlignment="Left" Margin="10,45,0,0" TextWrapping="Wrap" Text="Введіть дату одужання" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<TextBox x:Name="Nadb" HorizontalAlignment="Left" Margin="10,115,0,0" TextWrapping="Wrap" Text="Введіть надбавку" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<DataGrid x:Name="dt1" Height="382" Margin="205,0,10,0" VerticalAlignment="Center" />

<Button Content="Видалити" HorizontalAlignment="Left" Margin="10,255,0,0" VerticalAlignment="Top" Height="30" Width="170" Click="Button\_Click" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<TextBox x:Name ="IDField" HorizontalAlignment="Left" Margin="10,150,0,0" TextWrapping="Wrap" Text="Введіть ID" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<ComboBox x:Name="DelFrom" HorizontalAlignment="Left" Margin="10,185,0,0" VerticalAlignment="Top" Width="170" Height="30" >

<ComboBoxItem Content="SickID" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<ComboBoxItem Content="SalaryID" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

</ComboBox>

<Button x:Name ="Add" Content="Додати" HorizontalAlignment="Left" Margin="10,220,0,0" VerticalAlignment="Top" Width="170" Click="Add\_Click" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Повернутися" HorizontalAlignment="Left" Margin="10,360,0,0" VerticalAlignment="Top" Height="30" Width="170" Click="Return\_Click" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<ComboBox HorizontalAlignment="Left" Margin="10,325,0,0" VerticalAlignment="Top" Width="170" Height="30">

<Button Content="Кінцева зарплата" Click="FullSalary\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Оклад, стаж " Click="Salary\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Податки " Click="Tax\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Сімейний статус " Click="MarSt\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Лікарняні " Click="Sick\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

<Button Content="Премії " Click="Prem\_Click" Background="DodgerBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

</ComboBox>

<Label Content="Оберіть данні" HorizontalAlignment="Left" Margin="10,290,0,0" VerticalAlignment="Top" Width="170" Height="30" Background="SteelBlue" BorderBrush="WhiteSmoke" FontSize="13" Foreground="White"/>

</Grid>

</Window>