Міністерство освіти і науки України
Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

КУРСОВА РОБОТА
з дисципліни «Бази даних»
(назва дисципліни)
на тему: база даних для підтримки діловодства організації
Студента 2 курсу групи ІТ-03
спеціальності 121 «Інженерія програмного забезпечення»
Чабан А. Є.
(прізвище та ініціали)
Керівник:

доцент Ліщук К. І.
(посада, вчене звання, науковий ступінь, прізвище та ініціали)
Національна шкала _____
Кількість балів: _____ Оцінка ECTS _____
Члени комісії _____

_____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____ _____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)

_____ _____
(підпис) (вчене звання, науковий ступінь, прізвище та ініціали)
Київ – 2021 рік

Додаток Б

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет Інформатики та обчислювальної техніки
(повна назва)

Кафедра Інформатики та програмної інженерії
(повна назва)
Дисципліна Бази даних

Курс 2 Група ІТ-03 Семестр 1

З А В Д А Н Н Я
НА КУРСОВУ РОБОТУ СТУДЕНТУ

Чабан Антон Євгенович
(прізвище, ім'я, по батькові)

1. Тема роботи База даних для підтримки діловодства організації

керівник роботи Ліщук К. І.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

2. Строк подання студентом роботи: до 24.12.2021
3. Вихідні дані до роботи: розробити базу даних для підтримки діловодства організації

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити):
1) Вивчення літератури
2) Аналіз предметного середовища
3) Побудова ER-діаграми
4) Побудова реляційної схеми
5) Створення бази даних
6) Створення користувачів бази даних
7) Імпорт даних з використанням засобів СУБД
8) Створення запитів до розробленої БД
9) Оптимізація роботи запитів (за необхідності)

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)

6. Дата видачі завдання 31 жовтня 2021

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання курсового проекту	Строк виконання етапів проекту	Примітка
1	Вивчення літератури	До 05.11.2021	
2	Аналіз предметного середовища	До 12.11.2021	
3	Побудова ER-діаграми	До 28.11.2021	
4	Побудова реляційної схеми	До 28.11.2021	
5	Створення бази даних	До 30.11.2021	
6	Створення користувачів бази даних	До 30.11.2021	
7	Імпорт даних з використанням засобів СУБД	До 05.12.2021	
8	Створення запитів до розробленої БД	До 05.12.2021	
9	Оптимізація роботи запитів (за необхідності)	До 12.12.2021	
10	Написання звіту	До 20.12.2021	

Студент _____ Чабан А.Є.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Ліщук К.І.
(підпис) (прізвище та ініціали)

ЗМІСТ

ВСТУП

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналіз предметної області

1.1.1 Опис предметної області

1.1.2 Опис вхідних даних

1.1.3 Опис вихідних даних

1.2 Проектування бази даних

1.2.1 Інфологічна модель бази даних

1.2.1.1 Опис сутностей

1.2.1.2 Опис атрибутів

1.2.1.3 Опис зв'язків

1.2.1.4 ER-діаграма

1.2.2 Нормалізація таблиць при проектуванні бази даних.

1.2.3 Даталогічна модель бази даних

2 ПРАКТИЧНА ЧАСТИНА

2.1 Створення бази даних за допомогою MS SQL Server

2.1.1 Створення бази даних

2.1.2 Створення таблиць бази даних

2.1.3 Створення діаграми бази даних

2.1.4 Заповнення таблиць бази даних даними

2.1.5 Створення тригерів

2.1.6 Створення представлень

2.1.7 Створення функцій

2.2.1 DML запити

ВИСНОВКИ

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

ДОДАТКИ

Додаток А. Використані запити

Додаток Б. Заповнення таблиць реляційної бази даних

Вступ

Будь-яка БД є частиною інформаційної системи і призначена для збору, зберігання і маніпулювання інформацією.

У сучасному світі будь-яка організація працює з документами які надходять як ззовні, так і з середини організації. База даних являє собою спосіб управління інформації та її збереження для нормального функціонування організації.

База даних систематизує такі матеріали як інформацію про співробітників, документи та їх авторів (внутрішніх та зовнішніх), кар’єрні зміни та зміни та перебіг статусів документів.

Створюючи базу даних, ми прагнемо впорядкувати інформацію за різними ознаками і швидко витягати вибірку з поєднанням ознак.

Проектування бази даних являє собою складний трудомісткий процес відображення предметної області. Процес проектування бази даних полягає в створенні схеми бази даних і визначення обмежень інформації.

База даних може зберігати величезну кількість інформації, і швидко надавати її після введення запиту користувача. Ось чому бази даних дуже актуальні в нашому сучасному світі.

База даних - це організована структура, призначена для зберігання, зміни і обробки взаємозалежної інформації, переважно великих обсягів.

1 ТЕОРЕТИЧНА ЧАСТИНА

1.1 Аналіз предметної області

1.1.1 Опис предметної області

Інформаційна система відповідає за забезпечення інформаційних процесів, забезпечення створення, поширення, використання, збереження і знищення інформації. Сама інформаційна база складається з однієї або декількох баз даних.

Предметна область - частина реального світу, що описує інформаційна система в залежності від її призначення.

В даний час усі організації повинні мати доступ до інформації. Це дуже важливо, тому бази даних, які задовольняють потреби зі зберіганням й управлінням даних, мають неабияку цінність, бо допомагають людям в цій справі.

При розробці реляційної бази даних «База даних для підтримки діловодства організації» було проведено дослідження предметної області.

Отже, маємо організацію в якій працюють робітники, які мають посаду, відділ адресу, заробітну плату та премії . Адресі відповідає країна, місто, вулиця та поштовий індекс. Також маємо документи, які можуть надходити від зошнішніх або внутрішніх авторів, робітника який з ним працює та його стан.

1.1.2 Опис вхідних даних

Знаючи свою предметну область нам потрібно ввести дані, що призначені для введення, щоб наша база даних була потрібною.
При розробці реляційної бази даних «База даних для підтримки діловодства організації» були виділені наступні вхідні дані:

- Інформація про робітників
- Інформація про посади
- Інформація про відділи
- Інформація про адреси
- Інформація про міста
- Інформація про країни
- Інформація про документи
- Інформація про стани документів
- Інформація про типи документів
- Інформація про можливі переходи між станами
- Інформація про зміни в кар’єрі робітників
- Інформація про перебіг розгляду документа

1.1.3 Опис вихідних даних

Вихідні дані - повідомлення і результати, які видаються самою системою. Беруться з постійних даних. Вихідними даними для бази даних для підтримки діловодства організації є вихідні запити(табл.1-13):

Таблиця 1 – Довідник робітників

Назва поля	Тип поля	Опис поля
workerID	int	ID співробітника
workerName	nvarchar	Ім’я співробітника
workerSurname	nvarchar	Прізвище співробітника
workerSalary	float	Зарплата співробітника
workerBonuses	float	Бонуси співробітника
workerAdressID	int	ID адреси співробітника
workerCurrentRankID	int	ID рангу співробітника
workerCurrenntDepID	int	ID відділу співробітника
workerDate	date	Дата зайняття посади

Таблиця 2 – довідник посади

Назва поля	Тип поля	Опис поля
rankID	int	ID посади
rankName	nvarchar	Назва посади

Таблиця 3 – довідник відділи

Назва поля	Тип поля	Опис поля
depID	int	ID відділу
depName	nvarchar	Назва відділу

Таблиця 4 – довідник адреси

Назва поля	Тип поля	Опис поля
addressID	int	ID адреси
addressCityID	int	ID міста за адресою
addressStreet	nvarchar	Вулиця за адресою
addressCountryID	int	ID країни за адресою
addressZIP	nvarchar	Поштовий індекс за адресою

Таблиця 5 – довідник країни

Назва поля	Тип поля	Опис поля
countryID	int	ID країни
countryName	nvarchar	Назва країни

Таблиця 6 – довідник міста

Назва поля	Тип поля	Опис поля
cityID	int	ID міста
cityName	nvarchar	Назва міста

Таблиця 7 – довідник кар’єрних змін

Назва поля	Тип поля	Опис поля
careerID	int	ID кар’єрних змін
careerWorkerFullName	nvarchar	Повне ім’я співробітника що зазнав кар’єрних змін
careerRankID	int	ID рангу співробітника що зазнав кар’єрних змін
careerDepID	int	ID відділу співробітника що зазнав кар’єрних змін
careerStartDate	date	Дата зайняття/початку роботи співробітника що зазнав кар’єрних змін
careerEndDate	date	Дата звільнення/завершення роботи на старій посаді

Таблиця 8 – довідник авторів

Назва поля	Тип поля	Опис поля
authorID	int	ID автора
isInner	bit	Внутрішній/зовнішній автор
authorWorkerID	int	ID автора (якщо внутрішній)
authorOutName	nvarchar	Ім’я зовнішнього автора
authorOutSurname	nvarchar	Прізвище зовнішнього автора

authorOutAdressID	int	Адреса зовнішнього автора

Таблиця 9 – довідник типів документів

Назва поля	Тип поля	Опис поля
docTypeID	int	ID типу документа
docType	nvarchar	Тип документа

Таблиця 10 – довідник станів документів

Назва поля	Тип поля	Опис поля
docStateID	int	ID стану документа
docState	nvarchar	Стан документа

Таблиця 11 – довідник переходів між станами

Назва поля	Тип поля	Опис поля
state1ID	int	ID першого стану
state2ID	int	ID стану до якого буде здійснено перехід

Таблиця 12 – довідник документів

Назва поля	Тип поля	Опис поля
docID	int	ID документа
docRegNum	int	Реєстраційний номер документа
docDate	date	Дата оформлення документа
docTypeID	int	ID типу документа
docFile	binary	Файл документа
docAuthorID	int	ID автора документа
docTitle	nvarchar	Заголовок документа
docDescription	nvarchar	Опис документа
docCurStateID	int	Стан документа

Таблиця 13 – довідник перебігу змін в документі

Назва поля	Тип поля	Опис поля
workflowID	int	ID перебігу змін в документі
workflowDocID	int	ID документу
workflowWorkerID	int	ID робітника який працює з документом
workflowDateStart	date	Дата початку роботи з документом
workflowDateEnd	date	Дата закінчення роботи з документом
workflowPrevStateID	int	Попердній стан документа
workflowNewStateID	int	Новий стан документа

1.2 Проектування бази даних

1.2.1 Інфологічна модель бази даних

Проектувальники інфологічної моделі розглядають як модель сутність - зв'язок”.
Призначення інфографічної моделі полягає у наданні найбільш дійсних способів збору та представлення інформації, що зберігається у базі даних. Ось чому інфологічну модель даних пробують будувати по аналогії з дійсною мовою.
Початковим етап в проектуванні баз даних - це інфологічна модель, яка допомагає в побудові таблиць та зв’язків між ними.
Існують різні способи опису інфологічної моделі, однак на даний момент найбільш популярним з підходів є той, що заснований на діаграмі "сутністьзв'язок". Отже, при створенні інфологічної моделі ми будемо використовувати ER

– діаграму. Ця інфологічна модель - це модель предметної області, яка визначає
сутності та зв'язки бази даних. Побудова такої моделі є індивідуальною і особливою, тому однієї методики створення інфологічної моделі немає, кожен
робить так як хоче, так як саме він бачить цю модель предметної області.
Інфологічна модель – це опис бази даних, яка буде існувати в майбутньому, опис бд поданий за допомогою діаграм, таблиць, дійсної мови. Будування цієї
моделі є продовженням аналізу предметної області інфологічна модель є представленням БД з точки зору того, хто розробляє її.
Головними конструктивними елементами інфологічної моделі даних є: сутності, їх атрибутів та типи зв’язків.

1.2.1.1 Опис сутностей

Одним із елементом інфологічної моделі "сутність-зв'язок" є сутності. Сутність - це те, про що накопичується інформація в інформаційній системі і що може бути однозначно унікальне ідентифіковане. При цьому ім'я сутності
повинно відображати клас об'єкта або тип об’єкта.

У відповідності з описом предметної області були отримано такі сутності:

- «Співробітники» - зберігається інформація про співробітників
- «Адреси» - зберігається інформація про адреси
- «Міста» - зберігається інформація про міста
- «Країни» - зберігається інформація про країни
- «Посади» - зберігається інформація про посади
- «Відділи» - зберігається інформація про відділи
- «Кар’єрні зміни» - зберігається інформація про кар’єрні зміни
- «Автори» - зберігається інформація про авторів
- «Типи документів» - зберігається інформація про типи документів
- «Стани документів» - зберігається інформація про стани документів
- «Переходи між станами» - зберігається інформація про можливі переходи між станами
- «Документи» - зберігається інформація про документи

- «Перебіг змін в документах» - зберігається інформація про перебіг змін в документах

1.2.1.2 Опис атрибутів

Атрибут - це поіменована характеристика сутності, за допомогою якої моделюється її властивість. Атрибути називають ще інформаційними елементами.

Сутність і екземпляр сутності можуть бути визначені наступним чином:

Сутність: Співробітник (workerID
workerName
workerSurname
workerSalary
workerBonuses
workerAdressID
workerCurrentRankID
workerCurrenntDepID
workerDate)

Екземпляр сутності:
4, Anton,Chaban, 50000, 17650, 8, 1, 2, 2019-11-10

У відповідності з описом предметної області були виділені наступні атрибути у кожній сутності:

Працівники:

workerID
workerName
workerSurname
workerSalary
workerBonuses
workerAdressID
workerCurrentRankID
workerCurrenntDepID
workerDate

Посади:

rankID
rankName

Відділи:

depID
depName

Адреси:

addressID
addressCityID
addressStreet
addressCountryID
addressZIP

Країни:

countryID
countryName

Міста:

cityID
cityName

Кар’єрні зміни:

careerID
careerWorkerFullName
careerRankID
careerDepID
careerStartDate
careerEndDate

Автори:

authorID
isInner
authorWorkerID
authorOutName
authorOutSurname
authorOutAdressID

Типи документів:

docTypeID
docType

Стани документів:

docStateID
docState

Переходи станів:

state1ID
state2ID

Документи:

docID
docRegNum
docDate
docTypeID
docFile
docAuthorID
docTitle
docDescription
docCurStateID

Перебіг змін в документі:

workflowID
workflowDocID
workflowWorkerID
workflowDateStart
workflowDateEnd
workflowPrevStateID
workflowNewStateID

1.2.1.3 Опис зв'язків

Дві сутності можуть пов'язуватися через зв'язок екземплярів однієї сутності з екземплярами іншої сутності.

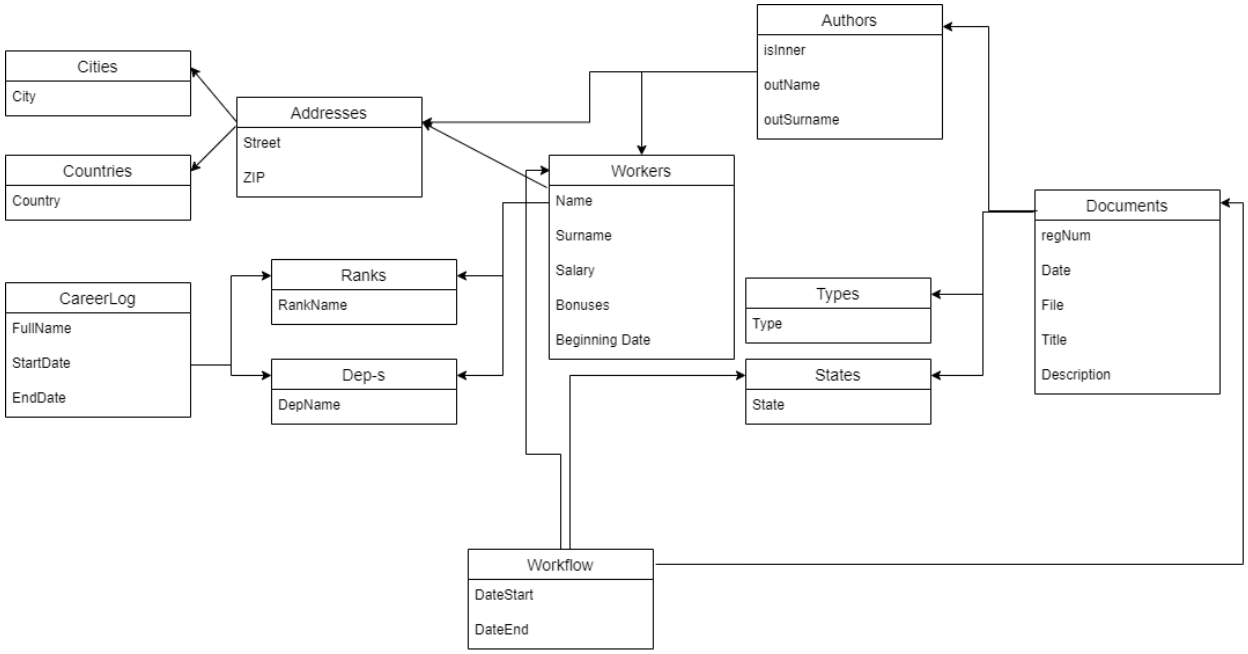
Основна вимога бази даних - вміти знаходити одну сутність за значеннями інших, для чого необхідно встановити зв’язок між ними.
Так як часто в базах даних створюють більше 50 сутностей, то між цими сутністями може бути дуже багато зв’язків. Складність інфологічної моделей визначається наявністю великої кількості зв’язків.
За допомогою зв'язків показують відношення між сутністю та атрибутами.
Зв’язкам можна надати імена для зручності. Термін зв’язок використовується для позначення типу зв’язку і може відображати одноразовість чи багаторазовість.
За направленістю розрізняють такі зв'язки:
1) Однонаправлений зв'язок поділяється на:
о однозначний, якщо один екземпляр однієї сутності відповідає точно одному екземпляру іншої сутності, зворотного зв'язку не надається;
о багатозначний, якщо екземпляр сутності відповідає декільком екземплярам іншої сутності, зворотного зв'язку теж не надається. [6]
2) Двонаправлений зв'язок поділяється на:
о однозначний "один до одного" (1:1) одному екземпляру сутності однієї відповідає лише одному екземпляру другої сутності (навпаки теж працює);
о двозначні: · "один до багатьох" (1: N) один екземпляр сутності відповідає декільком екземплярам іншої сутності, а кілька екземплярів іншої сутності відповідають лише одному екземпляру сутності.
· "багато до багатьох" (M: N) певна кількість екземплярів однієї сутності відповідає певній кількості екземплярів іншої сутності (навпаки теж працює).

В курсовій роботі були використані наступні типи зв'язків між таблицями реляційної бази даних «База даних для підтримки діловодства організації»:

Номер зв'язку	Головна таблиця	Дочірня таблиця	Тип зв'язку
1	Workers	Ranks	1:M
2	Workers	Departments	1:M
3	Workers	Addresses	1:M
4	Addresses	Countries	1:M
5	Addresses	Cities	1:M
6	Authors	Workers	1:M
7	Documents	Authors	1:M
8	Documents	DocTypes	1:M
9	Documents	DocStates	1:M
10	DocStates	PossibleStates	1:M
11	DocWorkFlow	Documents	1:M
12	DocWorkFlow	DocStates	1:M
13	DocWorkFlow	Workers	1:M

1.2.1.4 ER-діаграма

На рисунку представлена ER-діаграма бази даних «База даних для підтримки діловодства організації», на якій відображені всі сутності, їх атрибути та зв’язки між сутностями:



1.2.2 Нормалізація таблиць при проектуванні бази даних

Нормалізація – це формальна процедура, у ході якої атрибути даних групуються в таблиці, а таблиці групуються в базу даних.
Нормалізація бази даних – це одна з важливих вимог БД. Нормалізація потрібна для усунення надмірності даних, тобто видалення повторення даних в різних рядках однієї таблиці.

Перший крок до нормалізації бази даних - це позбавлення від повторень.
Другий крок до нормалізації бази даних – усі поля повинні бути неділимими. Тобто треба розділити усі поля, які можна, щоб в подальшому можна було легше шукати потрібну інформацію і працювати з БД.

Третій крок до нормалізації БД – не повинно бути полів, які можна знайти за допомогою інших полів.

Четвертий крок до нормалізації БД – не потрібні поля в БД, які позначають різні види одного і того ж.

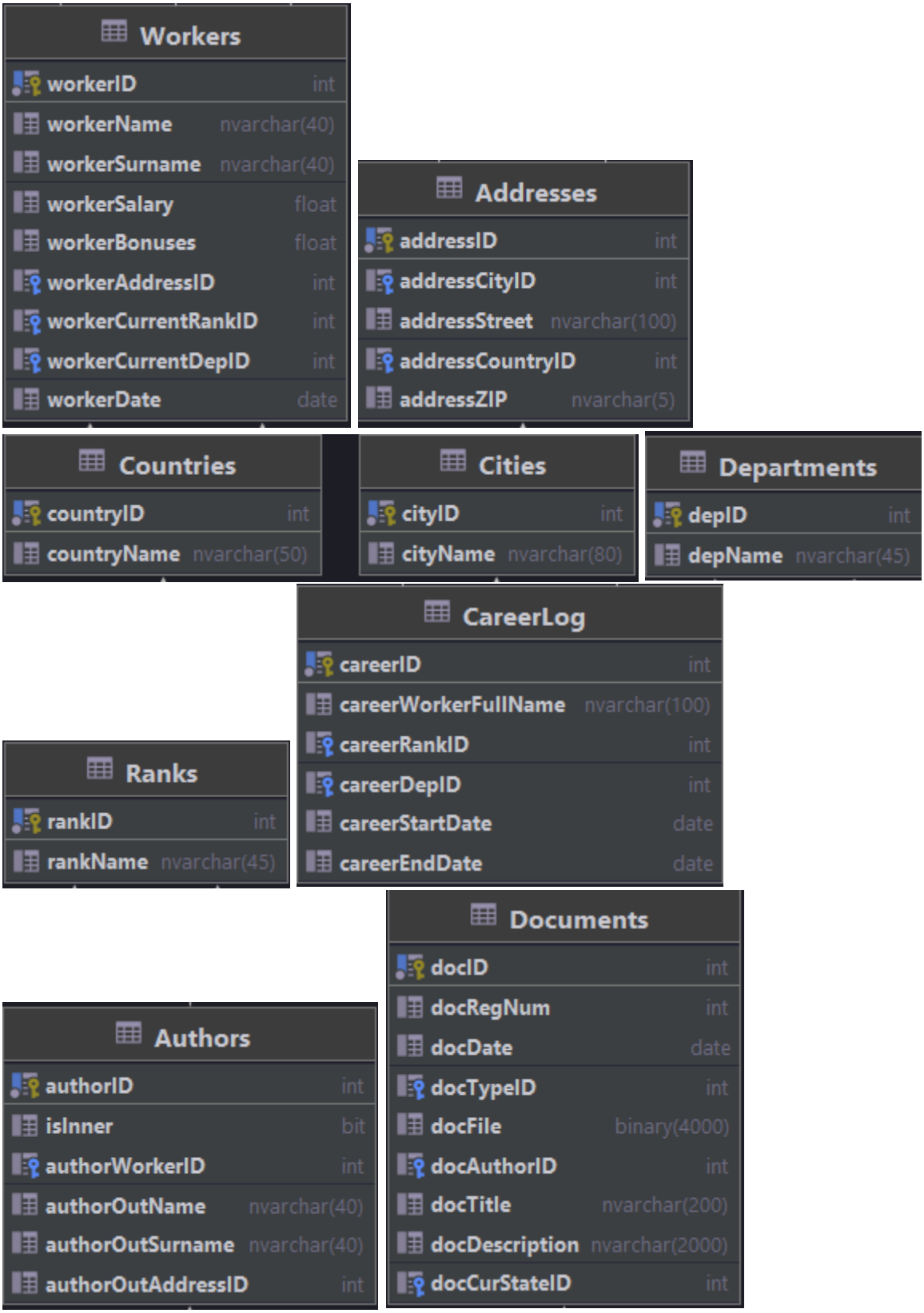
Кожна наступна нормальна форма має кращі властивості, ніж попередня форма. У теорії реляційних БД виділяють п’ять нормальних форми і нормальну форму Бойса-Кодда. Ми будемо розглядати тільки три нормальні форми.

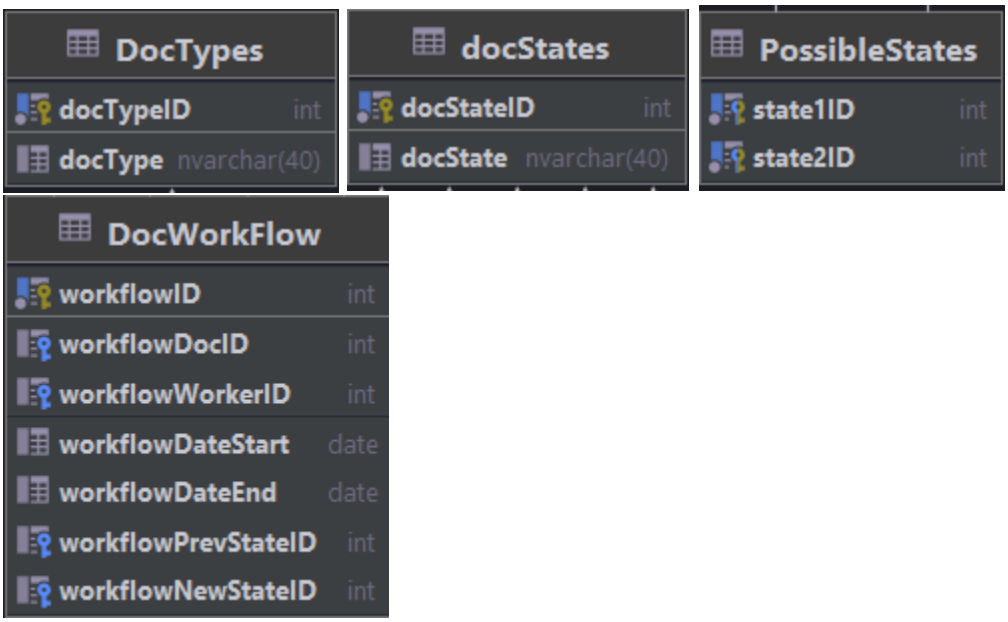
§ Перша нормальна форма вимагає, щоб кожне поле таблиці бази даних було неділимим і не містило повторень.

§ Друга нормальна форма вимагає, щоб всі атрибути таблиці залежали від первинного ключа.

§ Третя нормальна форма вимагає, щоб в таблиці не малося транзитивних залежностей між атрибутами таблиці.

Вигляд після нормалізації таблиць:





1.2.3 Даталогічна модель бази даних

На початку проектування бази даних використовують інфологічні моделі даних. Це потрібно для того, щоб описати саму структуру даних тоді, коли розробляється додаток, а даталогічні моделі бази даних деякими системами управління бд підтримуються.

Даталогічне проектування – затвердження рішення про те, як логічна модель буде реалізована у базі даних, що створюється з використанням обраної СУБД.

Логічна структура бази даних, а також сама база даних, яка вже заповнена даними, є відображенням певної предметної області. Специфіка предметної області, яка відбита в інфологічній моделі, впливає на вибір проектних рішень.

Даталогічна модель є моделлю бази даних логічного рівня і являє собою відображення логічних зв'язків між елементами даних до їхнього змісту й середовищу зберігання.

У результаті даталогічного проектування ми можемо отримати декілька варіантів логічної моделі. Ось чому важливо вірно оцінити отримані моделі і вибрати більш оптимальний варіант.

Також потрібно перевіряти умови виконання всіх запитів і вимог користувачів, й якщо щось не так, потрібно перебудувати модель даних. Для кожного поля таблиці реляційної бази даних «База даних для підтримки діловодства організації» вказується розмір поля, тип даних та обмеження для кожного поля, якщо необхідно та найменування самих атрибутів.

Працівники

Найменування атрибутів	Тип поля	Розмір поля
workerID	int	n
workerName	nvarchar	40
workerSurname	nvarchar	40
workerSalary	float	n
workerBonuses	float	n
workerAdressID	int	n
workerCurrentRankID	int	n
workerCurrenntDepID	int	n
workerDate	date	N

Посади

Найменування атрибутів	Тип поля	Розмір поля
rankID	int	n
rankName	nvarchar	45

Відділи

Найменування атрибутів	Тип поля	Розмір поля
depID	int	n
depName	nvarchar	45

Адреси

Найменування атрибутів	Тип поля	Розмір поля
addressID	int	n
addressCityID	int	n
addressStreet	nvarchar	100
addressCountryID	int	n
addressZIP	nvarchar	5

Країни

Найменування атрибутів	Тип поля	Розмір поля
countryID	int	n
countryName	nvarchar	50

Міста

Найменування атрибутів	Тип поля	Розмір поля
cityID	int	n
cityName	nvarchar	80

Кар’єрний лог

Найменування атрибутів	Тип поля	Розмір поля
careerID	int	n
careerWorkerFullName	nvarchar	100
careerRankID	int	n
careerDepID	int	n
careerStartDate	date	n
careerEndDate	date	n

Автори

Найменування атрибутів	Тип поля	Розмір поля
authorID	int	n
isInner	bit	n
authorWorkerID	int	n
authorOutName	nvarchar	40
authorOutSurname	nvarchar	40
authorOutAdressID	int	n

Типи документів

Найменування атрибутів	Тип поля	Розмір поля
docTypeID	int	n
docType	nvarchar	40

Стани документів

Найменування атрибутів	Тип поля	Розмір поля
docStateID	int	n
docState	nvarchar	40

Переходи між станами

Найменування атрибутів	Тип поля	Розмір поля
state1ID	int	n
state2ID	int	n

Документи

Найменування атрибутів	Тип поля	Розмір поля
docID	int	n
docRegNum	int	n
docDate	date	n
docTypeID	int	n
docFile	binary	4000
docAuthorID	int	n
docTitle	nvarchar	200
docDescription	nvarchar	2000
docCurStateID	int	n

Перебіг змін в документі

Найменування атрибутів	Тип поля	Розмір поля
workflowID	int	n
workflowDocID	int	n
workflowWorkerID	int	n
workflowDateStart	date	n
workflowDateEnd	date	n
workflowPrevStateID	int	n
workflowNewStateID	int	n

2 Практична частина

2.1 Створення бази даних за допомогою *DataGrip*

2.1.1 Створення бази даних

Далі за допомогою *sql*-команд створюємо реляційну базу даних із назвою docFlowDB та схему docFlowSc.

Також створимо користувачів для роботи з нашою БД.

```
CREATE LOGIN viewer WITH PASSWORD ='123'
GO
USE [docFlowDB]
GO
CREATE USER [viewer] FOR LOGIN [viewer]
GO
USE [docFlowDB]
GO
ALTER ROLE [db_datareader] ADD MEMBER [viewer]
GO
```

```
CREATE LOGIN accountant WITH PASSWORD ='123'
GO
USE [docFlowDB]
GO
CREATE USER [accountant] FOR LOGIN [accountant]
GO
USE [docFlowDB]
GO
ALTER ROLE [db_datawriter] ADD MEMBER [accountant]
GO
```

Перший користувач має права лише для читання, другий користувач може виконувати CRUD операції з таблицями.

2.1.2 Створення таблиць бази даних

В базі даних створюємо наступні таблиці:

- «Співробітники»
- «Адреси»
- «Міста»
- «Країни»
- «Посади»
- «Відділи»
- «Кар’єрні зміни»
- «Автори»
- «Типи документів»
- «Стани документів»
- «Переходи між станами»
- «Документи»
- «Перебіг змін в документах»

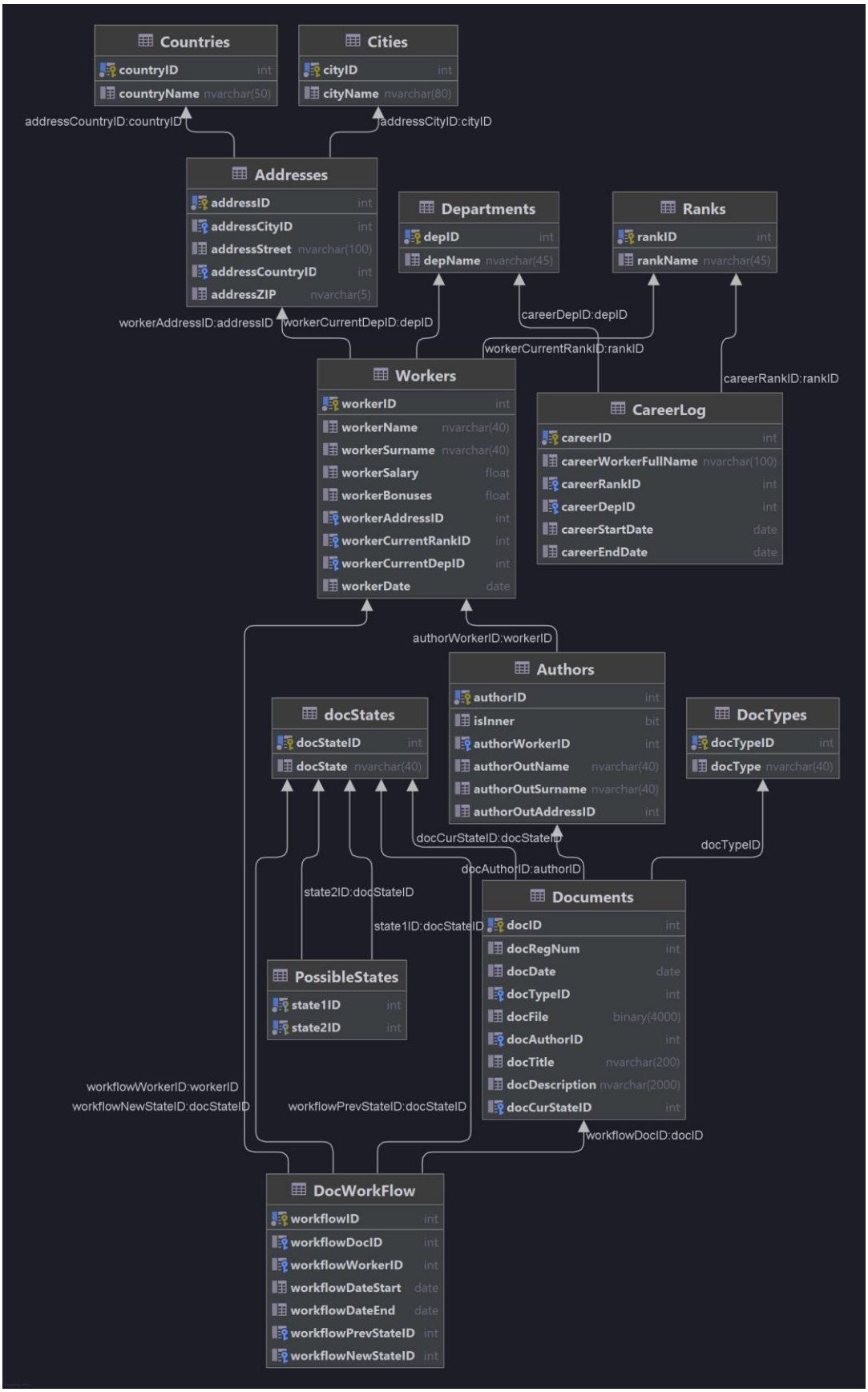
```
use docFlowDB
go

create table docFlowSc.Addresses
(
    addressID        int identity
        constraint Addresses_pk
            primary key nonclustered,
    addressCityID    int
        constraint Addresses_Cities_cityID_fk
            references docFlowSc.Cities,
    addressStreet    nvarchar(100),
    addressCountryID int
        constraint Addresses_Countries_countryID_fk
            references docFlowSc.Countries,
    addressZIP       nvarchar(5)
)
go

create unique index Addresses_addressID_uindex
    on docFlowSc.Addresses (addressID)
go
```

```sql
use docFlowDB
go

create table docFlowSc.Cities
(
    cityID   int identity
        constraint Cities_pk
            primary key nonclustered,
    cityName nvarchar(80)
)
go

create unique index Cities_cityID_uindex
    on docFlowSc.Cities (cityID)
go
```

Інші таблиці представлено в додатках.

2.1.3 Створення діаграми бази даних

У розділі діаграм в утиліті *DataGrip* створюємо нову діаграму, в яку додаємо зі списку тринадцять таблиць із даної предметної області та перевіряємо зв'язки між таблицями:


Countries
countryID int
countryName nvarchar(50)
Cities
cityID int
cityName nvarchar(80)
addressCountryID:countryID
addressCityID:cityID
Addresses
addressID int
addressCityID int
addressStreet nvarchar(100)
addressCountryID int
addressZIP nvarchar(5)
Departments
depID int
depName nvarchar(45)
Ranks
rankID int
rankName nvarchar(45)
workerAddressID:addressID
workerCurrentDepID:depID
careerDepID:depID
workerCurrentRankID:rankID
careerRankID:rankID
Workers
workerID int
workerName nvarchar(40)
workerSurname nvarchar(40)
workerSalary float
workerBonuses float
workerAddressID int
workerCurrentRankID int
workerCurrentDepID int
workerDate date
CareerLog
careerID int
careerWorkerFullName nvarchar(100)
careerRankID int
careerDepID int
careerStartDate date
careerEndDate date
authorWorkerID:workerID
Authors
authorID int
isInner bit
authorWorkerID int
authorOutName nvarchar(40)
authorOutSurname nvarchar(40)
authorOutAddressID int
docStates
docStateID int
docState nvarchar(40)
DocTypes
docTypeID int
docType nvarchar(40)
docCurStateID:docStateID
docTypeID
docAuthorID:authorID
state2ID:docStateID
state1ID:docStateID
Documents
docID int
docRegNum int
docDate date
docTypeID int
docFile binary(4000)
docAuthorID int
docTitle nvarchar(200)
docDescription nvarchar(2000)
docCurStateID int
PossibleStates
state1ID int
state2ID int
workflowWorkerID:workerID
workflowNewStateID:docStateID
workflowPrevStateID:docStateID
workflowDocID:docID
DocWorkFlow
workflowID int
workflowDocID int
workflowWorkerID int
workflowDateStart date
workflowDateEnd date
workflowPrevStateID int
workflowNewStateID int

2.1.4 Заповнення таблиць бази даних даними

Далі заповнимо наші таблиці даними для виконання запитів (використані запити в додатках)

```
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (1, N'Valeriy', N'Gusak', 23000, 1000,
41, 4, 9, N'2012-01-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (2, N'Viktor', N'Nosov', 11000, 200,
23, 11, 10, N'2005-05-17');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (3, N'Muhamed', N'Ali', 15000, 300,
32, 14, 2, N'1974-10-26');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (4, N'Anton', N'Chaban', 50000, 17650,
8, 1, 2, N'2019-11-10');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (5, N'Adam', N'Abraham', 25188, 216,
22, 3, 6, N'2021-12-11');

etc...
```

2.1.5 Створення тригерів

Створимо тригери для таблиці Workers та Documents

Які реагують після настання подій Insert Update Delete

Workers – тригер оновлює інформацію про робітника при виконанні одної з операцій. При Delete буде додана кінцева дата (сьогодняшня), при Insert в поле початку роботи додаємо сьогоднішню дату, при Update додаємо новий запис з новим рангом а дату залишаємо стару.

Documents – при доданні документу в лог додається інформація про його стан та дату початку роботи з документом. При оновленні – перевіряється чи є стан термінальним, якщо так – додається кінцева дата (сьогодняшня).

Створення тригерів в додатках.

2.1.6 Створення представлень

Маємо два представлення.

Представлення allAuthors показує всіх авторів та їх статус (внутрішній/зовнішній)

	FullName	AuthorStatus
1	Adrian Allan	Inner Author
2	Alexander Anderson	Inner Author
3	Anna Clark	Outer Author
4	Anton Chaban	Inner Author
5	Bob Abob	Outer Author
6	Joseph Flow	Outer Author
7	Matthew Ball	Inner Author
8	Stanislav Boiko	Outer Author
9	Valeriy Gusak	Inner Author
10	Viktor Toretto	Outer Author

Представлення documentsInfo показує інформацію про документи (авторів, їх статус, стан і реєстраційний номер)

	docRegNum	docType	AuthorName	docState
1	10000	Employee stock option agr…	Valeriy Gusak	Accepted
2	10001	Business reports	Anton Chaban	Archive
3	10002	Documentation of bylaws	Anton Chaban	Refused
4	10003	Minutes of business meeti…	Bob Abob	Approval
5	10004	Franchise agreement	Joseph Flow	Preparing
6	10005	Documentation of bylaws	Alexander Anderson	Refused
7	10006	Non-disclosure agreement	Anna Clark	Approval
8	10007	Financial documents	Viktor Toretto	Accepted
9	10008	Business pitch deck	Matthew Ball	Signing
10	10009	Minutes of business meeti…	Stanislav Boiko	Accepted
11	10014	Non-disclosure agreement	Anton Chaban	Preparing

2.1.7 Створення функцій

Маємо дві реалізовані функції.

Функція expCalculator повертає досвід працівника в роках, вираховуючи різницю в датах.

```
create function expCalculator(@WorkDate date)
    returns int
as
begin
    declare @ret int;
    select @ret = datediff(year, @WorkDate, getdate());
    return @ret;
end
```

Функція fullNamer повертає повне ім’я співробітника конкатинуючи його ім’я та прізвище.

```
create function fullNamer(@name nvarchar(40), @surname nvarchar(40))
    returns nvarchar(85)
as
begin
    return concat(@name, ' ', @surname);
end
```

2.2.1 DML запити

Документи від робітників компанії:

	docRegNum	workerFullName	docType	docTitle	docState
1	10000	Valeriy Gusak	Employee stock option agreement	Amazing	Accepted
2	10001	Anton Chaban	Business reports	Just read this	Archive
3	10002	Anton Chaban	Documentation of bylaws	Need to review	Refused
4	10005	Alexander Anderson	Documentation of bylaws	Look at this dude!	Refused
5	10008	Matthew Ball	Business pitch deck	Good idea, check it	Signing
6	10014	Anton Chaban	Non-disclosure agreement	New Document For Today	Preparing

Документи від зовнішніх представників:

	docRegNum	workerFullName	docType	docTitle	docState
1	10003	Bob Abob	Minutes of business meetings	Check it out	Approval
2	10004	Joseph Flow	Franchise agreement	Man look at this	Preparing
3	10006	Anna Clark	Non-disclosure agreement	Refuse this doc pls	Approval
4	10007	Viktor Toretto	Financial documents	Just ignore it	Accepted
5	10009	Stanislav Boiko	Minutes of business meetings	Bad idea, refuse it	Accepted

Працівники з України:

	<anonymous>	rankName	countryName	cityName	addressStreet
1	Anton Chaban	Chief Executive Officer	Ukraine	Kyiv	Mayakovskogo av.
2	Alexander Anderson	Clerk	Ukraine	Lviv	Ivana Franka st.
3	Oleksandr Horduz	Product Manager	Ukraine	Kyiv	Mayakovskogo av.
4	Charles Bower	Accountant	Ukraine	Lviv	Ivana Franka st.
5	Dan Cameron	Accountant	Ukraine	Zhytomyr	Maksyma Rylskogo av.
6	Edward Churchill	Clerk	Ukraine	Uzhhorod	Ivana Mazepy st.

Працівники з досвідом менше 10 років:

	FullName	WorkerExp	rankName	countryName	cityName
1	Valeriy Gusak	9	Chief Technology Officer	Eritrea	Epinay-sous-Senart
2	Anton Chaban	2	Chief Executive Officer	Ukraine	Kyiv
3	Adam Abraham	0	Chief Financial Officer	Sweden	Berane
4	Adrian Allan	0	Chief Operating Officer	Iceland	Amuru
5	Anthony Avery	2	Junior Developer	Eritrea	Epinay-sous-Senart
6	Blake Ball	0	Chief Human Resources Officer	Venezuela	Live Oak
7	Oleksandr Horduz	0	Product Manager	Ukraine	Kyiv
8	Charles Bower	0	Accountant	Ukraine	Lviv
9	Christopher Buckland	1	Middle Developer	Ireland	Bang Sao Thong
10	Edward Churchill	4	Clerk	Ukraine	Uzhhorod
11	Evan Clarkson	0	Accountant	Sweden	Berane
12	Ivan Bosker	0	Product Manager	United States	Texas City
13	Josh Bush	0	Chief Technology Officer	United States	Washington

Працівники з ЗП >15k і досвідом більше 10 років:

	FullName	WorkerExp	workerSalary	rankName	countryName	cityName
1	Valeriy Gusak	9	23000	Chief Technology Officer	Eritrea	Epinay-sous-Senart
2	Anton Chaban	2	50000	Chief Executive Officer	Ukraine	Kyiv
3	Adam Abraham	0	25188	Chief Financial Officer	Sweden	Berane
4	Adrian Allan	0	22569	Chief Operating Officer	Iceland	Amuru
5	Blake Ball	0	23755	Chief Human Resources Officer	Venezuela	Live Oak
6	Oleksandr Horduz	0	22729	Product Manager	Ukraine	Kyiv
7	Evan Clarkson	0	16863	Accountant	Sweden	Berane

Працівники з досвідом меншим ніж середній:

	FullName	WorkerExp	workerSalary	rankName	countryName	cityName
1	Valeriy Gusak	9	23000	Chief Technology Officer	Eritrea	Epinay-sous-Senart
2	Viktor Nosov	16	11000	Middle Developer	Ireland	Bang Sao Thong
3	Anton Chaban	2	50000	Chief Executive Officer	Ukraine	Kyiv
4	Adam Abraham	0	25188	Chief Financial Officer	Sweden	Berane
5	Adrian Allan	0	22569	Chief Operating Officer	Iceland	Amuru
6	Anthony Avery	2	13140	Junior Developer	Eritrea	Epinay-sous-Senart
7	Blake Ball	0	23755	Chief Human Resources Officer	Venezuela	Live Oak
8	Oleksandr Horduz	0	22729	Product Manager	Ukraine	Kyiv
9	Cameron Blake	13	27471	Chief Marketing Officer	Tuvalu	Yuyao
10	Charles Bower	0	3158	Accountant	Ukraine	Lviv
11	Christopher Buckland	1	8207	Middle Developer	Ireland	Bang Sao Thong
12	Dan Cameron	12	6412	Accountant	Ukraine	Zhytomyr
13	David Campbell	16	18513	Business Analytic	Costa Rica	Basse-Terre
14	Dominic Carr	13	7877	HR	Switzerland	Sarpang
15	Edward Churchill	4	4118	Clerk	Ukraine	Uzhhorod
16	Evan Clarkson	0	16863	Accountant	Sweden	Berane
17	Ivan Bosker	0	13333	Product Manager	United States	Texas City
18	Josh Bush	0	2349	Chief Technology Officer	United States	Washington

Працівники в країні проживання яких є буква «і» і починається на «U»

	FullName	rankName	countryName	cityName
1	Anton Chaban	Chief Executive Officer	Ukraine	Kyiv
2	Alexander Anderson	Clerk	Ukraine	Lviv
3	Oleksandr Horduz	Product Manager	Ukraine	Kyiv
4	Charles Bower	Accountant	Ukraine	Lviv
5	Dan Cameron	Accountant	Ukraine	Zhytomyr
6	Dylan Chapman	Senior Developer	United States	Washington
7	Edward Churchill	Clerk	Ukraine	Uzhhorod
8	Ivan Bosker	Product Manager	United States	Texas City
9	Josh Bush	Chief Technology Officer	United States	Washington

Працівники котрі працювали з документами:

	FullName	docTitle	docState
1	Anton Chaban	Bad idea, refuse it	Accepted
2	Anton Chaban	Amazing	Accepted
3	Anton Chaban	Check it out	Approval
4	Anton Chaban	Look at this dude!	Refused
5	Anton Chaban	Need to review	Refused
6	Anton Chaban	Good idea, check it	Signing
7	Anton Chaban	Bad idea, refuse it	Accepted
8	Anton Chaban	Need to review	Refused
9	Anton Chaban	Check it out	Approval
10	Anton Chaban	Man look at this	Preparing
11	Anton Chaban	Look at this dude!	Refused
12	Anton Chaban	Refuse this doc pls	Approval
13	Anton Chaban	Just ignore it	Accepted
14	Anton Chaban	Amazing	Accepted
15	Anton Chaban	Just read this	Archive
16	Anton Chaban	New Document For Today	Preparing

Робітники які були звільнені:

	careerWorkerFullName	depName	rankName	careerStartDate	careerEndDate
1	Ivan Bosker	Export	Product Manager	2021-12-11	2021-12-11

Документи в проміжку 2000-10-23 і сьогоднішньою датою:

	docRegNum	docType	docState	docDate
1	10000	Employee stock option agreement	Accepted	2021-12-17
2	10001	Business reports	Archive	2021-12-17
3	10002	Documentation of bylaws	Refused	2021-12-17
4	10003	Minutes of business meetings	Approval	2021-12-17
5	10004	Franchise agreement	Preparing	2021-12-17
6	10005	Documentation of bylaws	Refused	2021-12-17
7	10006	Non-disclosure agreement	Approval	2021-12-17
8	10007	Financial documents	Accepted	2021-12-17
9	10008	Business pitch deck	Signing	2021-12-17
10	10009	Minutes of business meetings	Accepted	2021-12-17
11	10014	Non-disclosure agreement	Preparing	2021-12-17

Топ 3 найвищі ЗП:

	<anonymous>	workerSalary	rankName	depName
1	Anton Chaban	50000	Chief Executive Officer	IT
2	Connor Butler	29000	Chief Operating Officer	Import
3	Austin Bailey	28008	Chief Financial Officer	Public Relations

Топ 5 найвищих премій:

	FullName	workerSalary	workerBonuses	rankName	depName
1	Anton Chaban	50000	17650	Chief Executive Officer	IT
2	Oleksandr Horduz	22729	9999	Product Manager	Procurement
3	Valeriy Gusak	23000	1000	Chief Technology Officer	HR
4	Alexander Anderson	3666	974	Clerk	Marketing
5	Benjamin Baker	7467	943	HR	Marketing

Кар’єрна історія Josh Bush:

	careerWorkerFullName	rankName	depName	careerStartDate
1	Josh Bush	Middle Developer	Public Relations	2021-12-10
2	Josh Bush	Senior Developer	Public Relations	2021-12-11
3	Josh Bush	Business Analytic	Public Relations	2021-12-12
4	Josh Bush	Chief Technology Officer	Public Relations	2021-12-12

Працівники чий поштовий індекс починається на 4:

	FullName	cityName	addressZIP
1	Adrian Allan	Amuru	49257
2	Alan Alsop	Moers	46167
3	Matthew Ball	Amuru	49257

Працівники з найбільшою кількістю документів:

	workerSurname	DocCount
1	Chaban	3
2	Gusak	1
3	Anderson	1
4	Ball	1

Документи з термінальними станами:

	docTitle	FullName	workflowDateStart	workflowDateEnd	docState
1	Amazing	Anton Chaban	2007-02-03	2021-12-11	Accepted
2	Look at this dude!	Anton Chaban	1974-08-22	2021-12-11	Refused
3	Need to review	Anton Chaban	2008-11-11	2021-12-12	Refused
4	Bad idea, refuse it	Anton Chaban	2021-12-11	2021-12-17	Accepted
5	Need to review	Anton Chaban	2021-12-12	2021-12-17	Refused
6	Look at this dude!	Anton Chaban	2021-12-11	2021-12-17	Refused
7	Just ignore it	Anton Chaban	1995-04-22	2021-12-17	Accepted
8	Amazing	Anton Chaban	2021-12-11	2021-12-17	Accepted
9	Just read this	Anton Chaban	2002-02-01	2021-12-17	Archive

Працівники які працюють з 2000 року:

	FullName	workerDate	cityName	countryName
1	Valeriy Gusak	2012-01-11	Epinay-sous-Senart	Eritrea
2	Viktor Nosov	2005-05-17	Bang Sao Thong	Ireland
3	Anton Chaban	2019-11-10	Kyiv	Ukraine
4	Adam Abraham	2021-12-11	Berane	Sweden
5	Adrian Allan	2021-12-11	Amuru	Iceland
6	Anthony Avery	2019-02-23	Epinay-sous-Senart	Eritrea
7	Blake Ball	2021-01-05	Live Oak	Venezuela
8	Matthew Ball	2001-08-08	Amuru	Iceland
9	Oleksandr Horduz	2021-12-12	Kyiv	Ukraine
10	Cameron Blake	2008-10-18	Yuyao	Tuvalu
11	Charles Bower	2021-12-11	Lviv	Ukraine
12	Christopher Buckland	2020-06-08	Bang Sao Thong	Ireland
13	Dan Cameron	2009-04-11	Zhytomyr	Ukraine
14	David Campbell	2005-05-05	Basse-Terre	Costa Rica
15	Dominic Carr	2008-12-05	Sarpang	Switzerland
16	Edward Churchill	2017-06-17	Uzhhorod	Ukraine
17	Evan Clarkson	2021-12-11	Berane	Sweden
18	Ivan Bosker	2021-12-11	Texas City	United States
19	Josh Bush	2021-12-12	Washington	United States

Документи від зовнішніх авторів в описі яких фігурує слово:

	docRegNum	docDate	authorOutSurname	docDescription
1	10003	2021-12-17	Abob	Lorem ipsum dolor sit amet, consectetur adipiscing elit, se…
2	10004	2021-12-17	Flow	Lorem ipsum dolor sit amet, consectetur adipiscing elit, se…
3	10006	2021-12-17	Clark	Lorem ipsum dolor sit amet, consectetur adipiscing elit, se…
4	10009	2021-12-17	Boiko	Lorem ipsum dolor sit amet, consectetur adipiscing elit, se…

Union внуштрішніх та зовнішніх авторів:

	FullName
1	Adrian Allan
2	Alexander Anderson
3	Anna Clark
4	Anton Chaban
5	Bob Abob
6	Joseph Flow
7	Matthew Ball
8	Stanislav Boiko
9	Valeriy Gusak
10	Viktor Toretto

Адреси в яких указана вулиця:

	cityName	countryName	addressZIP	addressStreet
1	Kyiv	Ukraine	22321	Mayakovskogo av.
2	Kharkiv	Ukraine	33333	Shevchenka st.
3	Lviv	Ukraine	14880	Ivana Franka st.
4	Uzhhorod	Ukraine	13370	Ivana Mazepy st.
5	Zhytomyr	Ukraine	22800	Maksyma Rylskogo av.

Порівняльна характеристика пошуку в таблиці з використанням індексу, та без:

```
docFlowDB> select * from docFlowSc.Cities where cityName = 'Rajmahal'
[2021-12-18 20:07:58] 1 row retrieved starting from 1 in 49 ms (execution: 14 ms, fetching: 35 ms)
docFlowDB> CREATE INDEX index1 ON docFlowSc.Cities (cityName)
[2021-12-18 20:08:06] completed in 52 ms
docFlowDB> select * from docFlowSc.Cities where cityName = 'Rajmahal'
[2021-12-18 20:08:10] 1 row retrieved starting from 1 in 24 ms (execution: 5 ms, fetching: 19 ms)
```

ВИСНОВКИ

Метою курсової роботи було проектування бази даних діловодства організації. Для виконання курсової роботи були проведені необхідні дослідження, що стосуються структури організації, та перебігу розгляду документів. Спочатку ми провели повний аналіз нашої предметної області і тоді вже почали її описувати.

Після цього була побудована концептуальна модель для якої ми використовували мову ER-опису предметної області, яка базується на концепції, що інформаційна модель предметної області може бути описана із застосування таких понять, як сутність, атрибут, зв’язок. Після побудови ER діаграми, ми провели нормалізацію таблиць нашої бази даних. І ми створили даталогічну модель бази даних. І тільки після цих завдань ми приступили к практичній частині і почали створювати саму базу даних, її таблиць, потім створили діаграму, щоб перевірити цілісність нашої бази даних і коректність. Далі ми заповняли таблиці бази даних, створювали тригери, представлення, функції та застосовували наші знання щодо вибірки інформації.

В даній курсовій роботі була розроблена база даних «База даних для підтримки діловодства організації» в системі управління базами даних Microsoft SQL Server 2019 (RTM) - 15.0.2000.5 (X64).

Створення бази даних для підтримки діловодства організації є актуальною та користною темою, адже будь-яка сучасна організація працює як з внутрішніми, так і з зовнішніми документами, та має вести їх перелік для уникнення проблем з законами.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

Microsoft SQL documentation - https://docs.microsoft.com/en-us/sql/?view=sql-server-ver15

Quick start with DataGrip - https://www.jetbrains.com/help/datagrip/quick-start-with-datagrip.html

Лекційний матеріал з предмету «Бази данних»

Intro to SQL - https://www.kaggle.com/learn/intro-to-sql

Advanced SQL - https://www.kaggle.com/learn/advanced-sql

ДОДАТКИ

Додаток А. Використані запити

```
use docFlowDB
go

create table docFlowSc.Cities
(
    cityID   int identity
        constraint Cities_pk
            primary key nonclustered,
    cityName nvarchar(80)
)
go

create index index1
    on docFlowSc.Cities (cityName)
go

create table docFlowSc.Countries
(
    countryID   int identity
        constraint Countries_pk
            primary key nonclustered,
    countryName nvarchar(50)
)
go

create table docFlowSc.Addresses
(
    addressID        int identity
        constraint Addresses_pk
            primary key nonclustered,
    addressCityID    int
        constraint Addresses_Cities_cityID_fk
            references docFlowSc.Cities,
    addressStreet    nvarchar(100),
    addressCountryID int
        constraint Addresses_Countries_countryID_fk
            references docFlowSc.Countries,
    addressZIP       nvarchar(5)
)
go

create unique index Addresses_addressID_uindex
    on docFlowSc.Addresses (addressID)
go

create unique index Countries_countryID_uindex
    on docFlowSc.Countries (countryID)
go
```

```sql
create table docFlowSc.Departments
(
    depID   int identity
        constraint Departments_pk
            primary key nonclustered,
    depName nvarchar(45)
)
go

create unique index Departments_depID_uindex
    on docFlowSc.Departments (depID)
go

create table docFlowSc.DocTypes
(
    docTypeID int identity
        constraint DocTypes_pk
            primary key nonclustered,
    docType   nvarchar(40)
)
go

create unique index DocTypes_docTypeID_uindex
    on docFlowSc.DocTypes (docTypeID)
go

create table docFlowSc.Ranks
(
    rankID   int identity
        constraint Ranks_pk
            primary key nonclustered,
    rankName nvarchar(45)
)
go

create table docFlowSc.CareerLog
(
    careerID             int identity
        constraint CareerLog_pk
            primary key nonclustered,
    careerWorkerFullName nvarchar(100),
    careerRankID         int
        constraint CareerLog_Ranks_rankID_fk
            references docFlowSc.Ranks,
    careerDepID          int
        constraint CareerLog_Departments_depID_fk
            references docFlowSc.Departments,
    careerStartDate      date,
    careerEndDate        date
)
go

create unique index CareerLog_careerID_uindex
```

```sql
    on docFlowSc.CareerLog (careerID)
go

create unique index Ranks_rankID_uindex
    on docFlowSc.Ranks (rankID)
go

create table docFlowSc.Workers
(
    workerID            int identity
        constraint Workers_pk
            primary key nonclustered,
    workerName          nvarchar(40),
    workerSurname       nvarchar(40),
    workerSalary        float,
    workerBonuses       float,
    workerAddressID     int
        constraint Workers_Addresses_addressID_fk
            references docFlowSc.Addresses,
    workerCurrentRankID int
        constraint Workers_Ranks_rankID_fk
            references docFlowSc.Ranks,
    workerCurrentDepID  int
        constraint Workers_Departments_depID_fk
            references docFlowSc.Departments,
    workerDate          date
)
go

create table docFlowSc.Authors
(
    authorID           int identity
        constraint Authors_pk
            primary key nonclustered,
    isInner            bit,
    authorWorkerID     int
        constraint Authors_Workers_workerID_fk
            references docFlowSc.Workers,
    authorOutName      nvarchar(40),
    authorOutSurname   nvarchar(40),
    authorOutAddressID int
)
go

create unique index Authors_authorID_uindex
    on docFlowSc.Authors (authorID)
go

create unique index Workers_workerID_uindex
    on docFlowSc.Workers (workerID)
go

create trigger Workers_Insert_Update_Delete
```

```
    on docFlowSc.Workers
    after insert , update, delete
    as
begin
    DECLARE @Operation VARCHAR(6)

    IF EXISTS(SELECT 0 FROM inserted)
        BEGIN
            IF EXISTS(SELECT 0 FROM deleted)
                BEGIN
                    SELECT @Operation = 'UPDATE'
                END
            ELSE
                BEGIN
                    SELECT @Operation = 'INSERT'
                END
        END
    ELSE
        BEGIN
            SELECT @Operation = 'DELETE'
        END
    PRINT @Operation
    IF @Operation = 'DELETE'
        insert into docFlowSc.CareerLog (careerWorkerFullName, careerRankID,
careerDepID, careerStartDate,
                                         careerEndDate)
        select (select concat(d.workerName, ' ', d.workerSurname)),
               d.workerCurrentRankID,
               d.workerCurrentDepID,
               d.workerDate,
               getdate()
        from deleted d;

    IF @Operation = 'INSERT'
        insert into docFlowSc.CareerLog
        select (select concat(i.workerName, ' ', i.workerSurname)),
               i.workerCurrentRankID,
               i.workerCurrentDepID,
               getdate(),
               null
        from inserted i;

    IF @Operation = 'UPDATE'
        insert into docFlowSc.CareerLog (careerWorkerFullName, careerRankID,
careerDepID, careerStartDate,
                                         careerEndDate)
        select (select concat(i.workerName, ' ', i.workerSurname)),
               i.workerCurrentRankID,
               i.workerCurrentDepID,
               d.workerDate,
               null
        from docFlowSc.Workers w
                 inner join deleted d on w.workerID = d.workerID
```

```
                inner join inserted i on w.workerID = i.workerID;
    update docFlowSc.Workers
    set workerDate = getdate()
    where workerID = (select i.workerID
                      from docFlowSc.Workers w
                               inner join inserted i on w.workerID =
i.workerID)
end
go

create table docFlowSc.docStates
(
    docStateID int identity
        constraint docStates_pk
            primary key nonclustered,
    docState   nvarchar(40)
)
go

create table docFlowSc.Documents
(
    docID          int identity
        constraint Documents_pk
            primary key nonclustered,
    docRegNum      int,
    docDate        date,
    docTypeID      int
        constraint Documents_DocTypes_docTypeID_fk
            references docFlowSc.DocTypes,
    docFile        binary(4000),
    docAuthorID    int
        constraint Documents_Authors_authorID_fk
            references docFlowSc.Authors,
    docTitle       nvarchar(200),
    docDescription nvarchar(2000),
    docCurStateID  int
        constraint Documents_docStates_docStateID_fk
            references docFlowSc.docStates
)
go

create table docFlowSc.DocWorkFlow
(
    workflowID          int identity
        constraint DocWorkFlow_pk
            primary key nonclustered,
    workflowDocID       int
        constraint DocWorkFlow_Documents_docID_fk
            references docFlowSc.Documents,
    workflowWorkerID    int
        constraint DocWorkFlow_Workers_workerID_fk
            references docFlowSc.Workers,
    workflowDateStart   date,
```

```sql
    workflowDateEnd     date,
    workflowPrevStateID int
        constraint DocWorkFlow_docStates_docStateID_fk
            references docFlowSc.docStates,
    workflowNewStateID  int
        constraint DocWorkFlow_docStates_docStateID_fk_2
            references docFlowSc.docStates
)
go

create unique index DocWorkFlow_workflowID_uindex
    on docFlowSc.DocWorkFlow (workflowID)
go

create unique index Documents_docID_uindex
    on docFlowSc.Documents (docID)
go

create trigger DocWorkFlowChanges
    on docFlowSc.Documents
    after insert , update, delete
    as
begin
    DECLARE @Operation VARCHAR(6)

    IF EXISTS(SELECT 0 FROM inserted)
        BEGIN
            IF EXISTS(SELECT 0 FROM deleted)
                BEGIN
                    SELECT @Operation = 'UPDATE'
                END
            ELSE
                BEGIN
                    SELECT @Operation = 'INSERT'
                END
        END
    ELSE
        BEGIN
            SELECT @Operation = 'DELETE'
        END
    PRINT @Operation

    IF @Operation = 'INSERT'
        insert into docFlowSc.DocWorkFlow (workflowDocID, workflowWorkerID,
workflowDateStart, workflowDateEnd,
                                           workflowPrevStateID,
workflowNewStateID)
        select i.docID, 4, getdate(), null, null, i.docCurStateID
        from inserted i;


    IF @Operation = 'UPDATE'
```

```
        insert into docFlowSc.DocWorkFlow (workflowDocID, workflowWorkerID,
workflowDateStart, workflowDateEnd,
                                           workflowPrevStateID,
workflowNewStateID)
        select i.docID,
               4,
               d.docDate,
               (select IIF((i.docCurStateID = 4 or i.docCurStateID = 7 or
i.docCurStateID = 8),
                           getdate(), null)),
               d.docCurStateID,
               i.docCurStateID
        from docFlowSc.Documents doc
                 inner join deleted d on doc.docID = d.docID
                 inner join inserted i on doc.docID = i.docID;
    update docFlowSc.Documents
    set docDate = getdate()
    where docID = (select i.docID
                   from docFlowSc.Documents doc
                            inner join inserted i on doc.docID = i.docID)
end
go

create table docFlowSc.PossibleStates
(
    state1ID int not null
        constraint PossibleStates_docStates_docStateID_fk
            references docFlowSc.docStates,
    state2ID int not null
        constraint PossibleStates_docStates_docStateID_fk_2
            references docFlowSc.docStates,
    constraint PK_States
        primary key (state1ID, state2ID)
)
go

create unique index docStates_docStateID_uindex
    on docFlowSc.docStates (docStateID)
go


-- ###### TRIGGERS START ######

-- Trigger for worker career logging

create trigger Workers_Insert_Update_Delete
    on docFlowSc.Workers
    after insert , update, delete
    as
begin
    DECLARE @Operation VARCHAR(6)

    IF EXISTS(SELECT 0 FROM inserted)
```

```
        BEGIN
            IF EXISTS(SELECT 0 FROM deleted)
                BEGIN
                    SELECT @Operation = 'UPDATE'
                END
            ELSE
                BEGIN
                    SELECT @Operation = 'INSERT'
                END
        END
    ELSE
        BEGIN
            SELECT @Operation = 'DELETE'
        END
    PRINT @Operation
    IF @Operation = 'DELETE'
        insert into docFlowSc.CareerLog (careerWorkerFullName, careerRankID,
careerDepID, careerStartDate,
                                         careerEndDate)
        select (select concat(d.workerName, ' ', d.workerSurname)),
               d.workerCurrentRankID,
               d.workerCurrentDepID,
               d.workerDate,
               getdate()
        from deleted d;

    IF @Operation = 'INSERT'
        insert into docFlowSc.CareerLog
        select (select concat(i.workerName, ' ', i.workerSurname)),
               i.workerCurrentRankID,
               i.workerCurrentDepID,
               getdate(),
               null
        from inserted i;

    IF @Operation = 'UPDATE'
        insert into docFlowSc.CareerLog (careerWorkerFullName, careerRankID,
careerDepID, careerStartDate,
                                         careerEndDate)
        select (select concat(i.workerName, ' ', i.workerSurname)),
               i.workerCurrentRankID,
               i.workerCurrentDepID,
               d.workerDate,
               null
        from docFlowSc.Workers w
                 inner join deleted d on w.workerID = d.workerID
                 inner join inserted i on w.workerID = i.workerID;
    update docFlowSc.Workers
    set workerDate = getdate()
    where workerID = (select i.workerID
                      from docFlowSc.Workers w
                               inner join inserted i on w.workerID =
i.workerID)
```

```
end
GO


-- Trigger for document workflow logging


create trigger DocWorkFlowChanges
    on docFlowSc.Documents
    after insert , update, delete
    as
begin
    DECLARE @Operation VARCHAR(6)

    IF EXISTS(SELECT 0 FROM inserted)
        BEGIN
            IF EXISTS(SELECT 0 FROM deleted)
                BEGIN
                    SELECT @Operation = 'UPDATE'
                END
            ELSE
                BEGIN
                    SELECT @Operation = 'INSERT'
                END
        END
    ELSE
        BEGIN
            SELECT @Operation = 'DELETE'
        END
    PRINT @Operation

    IF @Operation = 'INSERT'
        insert into docFlowSc.DocWorkFlow (workflowDocID, workflowWorkerID,
workflowDateStart, workflowDateEnd,
                                           workflowPrevStateID,
workflowNewStateID)
        select i.docID, 4, getdate(), null, null, i.docCurStateID
        from inserted i;


    IF @Operation = 'UPDATE'
        insert into docFlowSc.DocWorkFlow (workflowDocID, workflowWorkerID,
workflowDateStart, workflowDateEnd,
                                           workflowPrevStateID,
workflowNewStateID)
        select i.docID,
               4,
               d.docDate,
               (select IIF((i.docCurStateID = 4 or i.docCurStateID = 7 or
i.docCurStateID = 8),
                           getdate(), null)),
               d.docCurStateID,
               i.docCurStateID
```

```sql
        from docFlowSc.Documents doc
                inner join deleted d on doc.docID = d.docID
                inner join inserted i on doc.docID = i.docID;
    update docFlowSc.Documents
    set docDate = getdate()
    where docID = (select i.docID
                   from docFlowSc.Documents doc
                           inner join inserted i on doc.docID = i.docID)
end
GO

drop trigger docFlowSc.DocWorkFlowChanges;

-- ###### TRIGGERS END ######


-- ###### FUNCTIONS START ######


-- Function to calculate worker exp on current rank

create function expCalculator(@WorkDate date)
    returns int
as
begin
    declare @ret int;
    select @ret = datediff(year, @WorkDate, getdate());
    return @ret;
end

-- Function to concatenate Name + Surname and get worker Full Name

create function fullNamer(@name nvarchar(40), @surname nvarchar(40))
    returns nvarchar(85)
as
begin
    return concat(@name, ' ', @surname);
end

-- ###### FUNCTIONS END ######

-- ###### DML QUERIES START ######


-- Documents from inner #1
select docRegNum,
       (select concat(W.workerName, ' ', W.workerSurname)) workerFullName,
       DT.docType,
       docTitle,
       dS.docState
from docFlowSc.Documents
         inner join docFlowSc.Authors A on A.authorID = Documents.docAuthorID
         inner join docFlowSc.Workers W on W.workerID = A.authorWorkerID
```

```
        inner join docFlowSc.DocTypes DT on DT.docTypeID =
Documents.docTypeID
        inner join docFlowSc.docStates dS on dS.docStateID =
Documents.docCurStateID


-- Docs from outer #2
select docRegNum,
       (select concat(A.authorOutName, ' ', A.authorOutSurname))
workerFullName,
       DT.docType,
       docTitle,
       dS.docState
from docFlowSc.Documents
        inner join docFlowSc.Authors A on A.authorID = Documents.docAuthorID
        inner join docFlowSc.DocTypes DT on DT.docTypeID =
Documents.docTypeID
        inner join docFlowSc.docStates dS on dS.docStateID =
Documents.docCurStateID
where isInner = 0;

-- Workers from Ukraine #3

select (select concat(W.workerName, ' ', W.workerSurname)),
       R2.rankName,
       C.countryName,
       C2.cityName,
       A.addressStreet
from docFlowSc.Workers W
        inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
        inner join docFlowSc.Countries C on C.countryID = A.addressCountryID
        inner join docFlowSc.Cities C2 on C2.cityID = A.addressCityID
        inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
where countryName = 'Ukraine';

-- Workers w/ exp < 10 #4

select dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       dbo.expCalculator(w.workerDate)              WorkerExp,
       R2.rankName,
       C.countryName,
       C2.cityName
from docFlowSc.Workers W
        inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
        inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
        inner join docFlowSc.Countries C on C.countryID = A.addressCountryID
        inner join docFlowSc.Cities C2 on C2.cityID = A.addressCityID
where dbo.expCalculator(w.workerDate) < 10;

-- Workers w/ salary >15k and exp <10yr #5

select dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       dbo.expCalculator(w.workerDate)              WorkerExp,
```

```sql
       W.workerSalary,
       R2.rankName,
       C.countryName,
       C2.cityName
from docFlowSc.Workers W
         inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
         inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
         inner join docFlowSc.Countries C on C.countryID = A.addressCountryID
         inner join docFlowSc.Cities C2 on C2.cityID = A.addressCityID
where dbo.expCalculator(w.workerDate) < 10
  and W.workerSalary > 15000;

-- Workers w/ exp less then avg exp #6

with seniors as (
    select avg(dbo.expCalculator(workerDate)) as avgexp
    from docFlowSc.Workers
)
select dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       dbo.expCalculator(w.workerDate)              WorkerExp,
       W.workerSalary,
       R2.rankName,
       C.countryName,
       C2.cityName
from docFlowSc.Workers W
         inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
         inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
         inner join docFlowSc.Countries C on C.countryID = A.addressCountryID
         inner join docFlowSc.Cities C2 on C2.cityID = A.addressCityID
where dbo.expCalculator(w.workerDate) < (select avgexp from seniors)

-- Workers which country have 'i' in the middle and begins w/ 'U' #7

select dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       R2.rankName,
       C.countryName,
       C2.cityName
from docFlowSc.Workers W
         inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
         inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
         inner join docFlowSc.Countries C on C.countryID = A.addressCountryID
         inner join docFlowSc.Cities C2 on C2.cityID = A.addressCityID
where countryName like '%i%'
  and countryName like 'U%';


-- Workers which worked w/ documents #8

select dbo.fullNamer(W.workerName, W.workerSurname) FullName, d.docTitle,
ds.docState
from docFlowSc.Workers W
         inner join docFlowSc.DocWorkFlow DWF on W.workerID =
DWF.workflowWorkerID
```

```sql
        inner join docFlowSc.Documents D on D.docID = DWF.workflowDocID
        inner join docFlowSc.docStates dS on dS.docStateID = D.docCurStateID

-- Workers which were fired #9

select careerWorkerFullName, D.depName, R2.rankName, careerStartDate,
careerEndDate
from docFlowSc.CareerLog
        inner join docFlowSc.Departments D on D.depID =
CareerLog.careerDepID
        inner join docFlowSc.Ranks R2 on R2.rankID = CareerLog.careerRankID
where careerEndDate is not null;

-- Documents w/ date between 2000-10-23 and cur date #10

select docRegNum, DT.docType, dS.docState, docDate
from docFlowSc.Documents
        inner join docFlowSc.DocTypes DT on DT.docTypeID =
Documents.docTypeID
        inner join docFlowSc.docStates dS on dS.docStateID =
Documents.docCurStateID
where docDate between '2000-10-23' and getdate();

-- Top 3 best salaries #11

select top 3 dbo.fullNamer(W.workerName, W.workerSurname), workerSalary,
R2.rankName, D.depName
from docFlowSc.Workers W
        inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
        inner join docFlowSc.Departments D on D.depID = W.workerCurrentDepID
order by workerSalary desc;

-- Top 5 best bonuses #12

select top 5 dbo.fullNamer(W.workerName, W.workerSurname) FullName,
            workerSalary,
            workerBonuses,
            R2.rankName,
            D.depName
from docFlowSc.Workers W
        inner join docFlowSc.Ranks R2 on R2.rankID = W.workerCurrentRankID
        inner join docFlowSc.Departments D on D.depID = W.workerCurrentDepID
order by workerBonuses desc;

-- Work history of Josh Bush #13

select careerWorkerFullName, R2.rankName, D.depName, careerStartDate
from docFlowSc.CareerLog
        inner join docFlowSc.Ranks R2 on R2.rankID = CareerLog.careerRankID
        inner join docFlowSc.Departments D on D.depID =
CareerLog.careerDepID
where careerWorkerFullName =
```

```sql
      (select dbo.fullNamer(W.workerName, W.workerSurname) from
docFlowSc.Workers W where workerID = 41)
order by careerStartDate asc;

-- Workers which ZIP Code begins w/ 4 #14

select dbo.fullNamer(W.workerName, W.workerSurname) FullName, C.cityName,
A.addressZIP
from docFlowSc.Workers W
         inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
         inner join docFlowSc.Cities C on C.cityID = A.addressCityID
where addressZIP like '4%';

-- Inner authors w/ most documents #15

select W.workerSurname, count(1) DocCount
from docFlowSc.Documents
         inner join docFlowSc.Authors A on A.authorID = Documents.docAuthorID
         inner join docFlowSc.Workers W on W.workerID = A.authorWorkerID
where isInner = 1
group by W.workerSurname
order by count(1) desc;

-- Closed docs and its states #16

select D.docTitle,
       dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       workflowDateStart,
       workflowDateEnd,
       dS.docState
from docFlowSc.DocWorkFlow
         inner join docFlowSc.docStates dS on dS.docStateID =
DocWorkFlow.workflowNewStateID
         inner join docFlowSc.Documents D on D.docID =
DocWorkFlow.workflowDocID
         inner join docFlowSc.Workers W on W.workerID =
DocWorkFlow.workflowWorkerID
where workflowDateEnd is not null;

-- Workers which works from 2000 year for today #17

select dbo.fullNamer(W.workerName, W.workerSurname) FullName, workerDate,
C.cityName, C2.countryName
from docFlowSc.Workers W
         inner join docFlowSc.Addresses A on A.addressID = W.workerAddressID
         inner join docFlowSc.Cities C on C.cityID = A.addressCityID
         inner join docFlowSc.Countries C2 on C2.countryID =
A.addressCountryID
where workerDate between '2000-01-01' and getdate();

-- #18 Docs from outer authors w/ word 'Donec' inside

select docRegNum, docDate, A.authorOutSurname, docDescription
```

```sql
from docFlowSc.Documents
         inner join docFlowSc.Authors A on A.authorID = Documents.docAuthorID
where docDescription like '%Donec%'
  and isInner = 0

-- Inner and outer authors union #19

select dbo.fullNamer(W.workerName, W.workerSurname) FullName
from docFlowSc.Authors
         inner join docFlowSc.Workers W on W.workerID =
Authors.authorWorkerID
union
select dbo.fullNamer(A.authorOutName, A.authorOutSurname) FullName
from docFlowSc.Authors A
where isInner = 0

-- Addresses where street is mentioned #20

select C.cityName, C2.countryName, addressZIP, addressStreet
from docFlowSc.Addresses
         inner join docFlowSc.Cities C on C.cityID = Addresses.addressCityID
         inner join docFlowSc.Countries C2 on C2.countryID =
Addresses.addressCountryID
where addressStreet is not null;


-- ###### DML QUERIES END ######


-- INDEX COMPARING

CREATE INDEX index1 ON docFlowSc.Cities (cityName);
drop index docFlowSc.Cities.index1;
select * from docFlowSc.Cities where cityName = 'Rajmahal';

-- END OF COMPARING


-- ###### VIEWS START ######


-- View which shows all authors and their status(inner/outer)


create view allAuthors
as
select dbo.fullNamer(W.workerName, W.workerSurname) FullName,
       (select IIF((isInner = 1),
                   'Inner Author', 'Outer Author')) AuthorStatus
from docFlowSc.Authors
         inner join docFlowSc.Workers W on W.workerID =
Authors.authorWorkerID
union
```

```sql
select dbo.fullNamer(A.authorOutName, A.authorOutSurname) FullName,
       (select IIF((isInner = 1),
                   'Inner Author', 'Outer Author'))       AuthorStatus
from docFlowSc.Authors A
where isInner = 0

select *
from allAuthors;

-- View which shows all documents and its authors, status, states and regnum

create view documentsInfo
as
select docRegNum,
       DT.docType,
       (select IIF((isInner = 1), dbo.fullNamer(W.workerName,
W.workerSurname)
                   , dbo.fullNamer(A.authorOutName, A.authorOutSurname)))
AuthorName,
       dS.docState
from docFlowSc.Documents
         inner join docFlowSc.DocTypes DT on DT.docTypeID =
Documents.docTypeID
         inner join docFlowSc.Authors A on A.authorID = Documents.docAuthorID
         left join docFlowSc.Workers W on W.workerID = A.authorWorkerID
         inner join docFlowSc.docStates dS on dS.docStateID =
Documents.docCurStateID

select *
from documentsInfo;

-- Creating user role for viewing only

CREATE LOGIN viewer WITH PASSWORD ='123'
GO
USE [docFlowDB]
GO
CREATE USER [viewer] FOR LOGIN [viewer]
GO
USE [docFlowDB]
GO
ALTER ROLE [db_datareader] ADD MEMBER [viewer]
GO

EXECUTE AS USER = 'viewer'

-- Queries to check permissions
SELECT SUSER_NAME(), USER_NAME();
SELECT *
FROM fn_my_permissions(null, 'database');

drop user viewer;
drop login viewer;
```

```
revert;

-- Accountant user (have access to CRUD tables)

CREATE LOGIN accountant WITH PASSWORD ='123'
GO
USE [docFlowDB]
GO
CREATE USER [accountant] FOR LOGIN [accountant]
GO
USE [docFlowDB]
GO
ALTER ROLE [db_datawriter] ADD MEMBER [accountant]
GO

EXECUTE AS USER = 'accountant'


-- ###### SEQUENCE START ######

CREATE SEQUENCE DocNum
    AS bigint
    START WITH 10010
    INCREMENT BY 1
    MINVALUE -9223372036854775808
    MAXVALUE 9223372036854775807
    CACHE
GO

select next value for DocNum;

SELECT current_value
FROM sys.sequences
WHERE name = 'DocNum';
SELECT *
FROM sys.sequences
WHERE name = 'DocNum';


INSERT INTO docFlowSc.Documents (docRegNum, docDate, docTypeID, docFile,
docAuthorID, docTitle, docDescription,
                                 docCurStateID)
VALUES (next value for DocNum, getdate(),3,null,2,'New Document For Today',
'ssdsdads fedFUVSfyu', 1);
```

Додаток Б. Заповнення таблиць реляційної бази даних

Дані таблиці Addresses:

```
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (8, 267, N'Mayakovskogo
av.', 212, N'22321');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (9, 514, N'Shevchenka st.', 212, N'33333');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (10, 1075, N'Ivana Franka st.', 212, N'14880');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (11, 4003, N'Ivana Mazepy st.', 212, N'13370');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (12, 2242, N'Maksyma Rylskogo av.', 212, N'22800');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (13, 24253, null, 215, N'12312');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (14, 3693, null, 215, N'65231');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (15, 3988, null, 215, N'78324');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (16, 895, null, 215, N'61212');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (17, 123, null, 186, N'13640');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (18, 653, null, 210, N'24856');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (19, 854, null, 49, N'69796');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (20, 3221, null, 108, N'17885');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (21, 8767, null, 205, N'29282');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (22, 8231, null, 196, N'89915');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (23, 6853, null, 99, N'14564');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (24, 7652, null, 53, N'28137');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID, addressStreet, addressCountryID, addressZIP) VALUES (25, 32131, null, 115, N'36100');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (26, 8764, null, 163,
N'39859');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (27, 6523, null, 115,
N'22111');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (28, 9316, null, 83,
N'81557');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (29, 9032, null, 24,
N'84118');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (30, 7654, null, 165,
N'53281');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (31, 8761, null, 167,
N'96477');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (32, 4512, null, 130,
N'94272');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (33, 9781, null, 197,
N'14539');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (34, 1488, null, 29,
N'45688');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (35, 7652, null, 219,
N'18200');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (36, 2711, null, 97,
N'90069');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (37, 9873, null, 94,
N'49257');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (38, 8541, null, 126,
N'48156');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (39, 14355, null, 116,
N'75198');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (40, 4444, null, 56,
N'46167');
INSERT INTO docFlowDB.docFlowSc.Addresses (addressID, addressCityID,
addressStreet, addressCountryID, addressZIP) VALUES (41, 7777, null, 64,
N'98787');
```

Дані таблиці Authors:

```sql
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (1, 1, 1, null, null, null);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (2, 1, 4, null, null, null);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (3, 1, 6, null, null, null);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (4, 0, null, N'Bob', N'Abob', 16);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (5, 0, null, N'Joseph', N'Flow', 34);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (6, 1, 8, null, null, null);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (7, 0, null, N'Anna', N'Clark', 13);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (8, 0, null, N'Viktor', N'Toretto', 9);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (9, 1, 14, null, null, null);
INSERT INTO docFlowDB.docFlowSc.Authors (authorID, isInner, authorWorkerID, authorOutName, authorOutSurname, authorOutAddressID) VALUES (10, 0, null, N'Stanislav', N'Boiko', 8);
```

Дані таблиці CareerLog:

```sql
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (45, N'Charles Bower', 8, 2, N'2000-07-03', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (46, N'Adrian Allan', 2, 8, N'2011-05-02', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (47, N'Adam Abraham', 3, 6, N'1997-10-01', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (48, N'Oleksandr Horduz', 14, 6, N'2021-12-10', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (49, N'Josh Bush', 11, 3, N'2021-12-10', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName, careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (50, N'Josh Bush', 12, 3, N'2021-12-11', null);
```

```
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (51, N'Josh
Bush', 13, 3, N'2021-12-12', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (52, N'Josh
Bush', 4, 3, N'2021-12-12', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (38, N'Ivan
Bosker', 6, 4, N'2021-12-11', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (39, N'Ivan
Bosker', 9, 4, N'2021-12-11', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (40, N'Ivan
Bosker', 6, 4, N'2021-12-11', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (41, N'Evan
Clarkson', 8, 8, N'2021-12-11', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (42, N'Ivan
Bosker', 14, 4, N'2021-12-11', null);
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (43, N'Ivan
Bosker', 14, 4, N'2021-12-11', N'2021-12-11');
INSERT INTO docFlowDB.docFlowSc.CareerLog (careerID, careerWorkerFullName,
careerRankID, careerDepID, careerStartDate, careerEndDate) VALUES (44, N'Ivan
Bosker', 14, 4, N'2021-12-11', null);
```

Дані таблиці Cities:

Дані таблиці Countries:

```
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (1,
N'Afghanistan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (2,
N'Albania ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (3,
N'Algeria ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (4,
N'American Samoa ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (5,
N'Andorra ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (6,
N'Angola ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (7,
N'Anguilla ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (8,
N'Antigua & Barbuda ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES (9,
N'Argentina ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(10, N'Armenia ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(11, N'Aruba ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(12, N'Australia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(13, N'Austria ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(14, N'Azerbaijan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(15, N'Bahamas, The ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(16, N'Bahrain ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(17, N'Bangladesh ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(18, N'Barbados ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(19, N'Belarus ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(20, N'Belgium ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(21, N'Belize ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(22, N'Benin ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(23, N'Bermuda ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(24, N'Bhutan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(25, N'Bolivia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(26, N'Bosnia & Herzegovina ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(27, N'Botswana ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(28, N'Brazil ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(29, N'British Virgin Is. ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(30, N'Brunei ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(31, N'Bulgaria ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(32, N'Burkina Faso ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(33, N'Burma ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(34, N'Burundi ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(35, N'Cambodia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(36, N'Cameroon ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(37, N'Canada ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(38, N'Cape Verde ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(39, N'Cayman Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(40, N'Central African Rep. ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(41, N'Chad ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(42, N'Chile ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(43, N'China ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(44, N'Colombia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(45, N'Comoros ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(46, N'Congo, Dem. Rep. ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(47, N'Congo, Repub. of the ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(48, N'Cook Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(49, N'Costa Rica ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(50, N'Cote d''Ivoire ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(51, N'Croatia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(52, N'Cuba ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(53, N'Cyprus ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(54, N'Czech Republic ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(55, N'Denmark ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(56, N'Djibouti ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(57, N'Dominica ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(58, N'Dominican Republic ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(59, N'East Timor ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(60, N'Ecuador ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(61, N'Egypt ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(62, N'El Salvador ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(63, N'Equatorial Guinea ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(64, N'Eritrea ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(65, N'Estonia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(66, N'Ethiopia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(67, N'Faroe Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(68, N'Fiji ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(69, N'Finland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(70, N'France ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(71, N'French Guiana ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(72, N'French Polynesia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(73, N'Gabon ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(74, N'Gambia, The ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(75, N'Gaza Strip ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(76, N'Georgia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(77, N'Germany ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(78, N'Ghana ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(79, N'Gibraltar ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(80, N'Greece ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(81, N'Greenland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(82, N'Grenada ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(83, N'Guadeloupe ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(84, N'Guam ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(85, N'Guatemala ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(86, N'Guernsey ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(87, N'Guinea ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(88, N'Guinea-Bissau ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(89, N'Guyana ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(90, N'Haiti ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(91, N'Honduras ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(92, N'Hong Kong ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(93, N'Hungary ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(94, N'Iceland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(95, N'India ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(96, N'Indonesia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(97, N'Iran ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(98, N'Iraq ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(99, N'Ireland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(100, N'Isle of Man ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(101, N'Israel ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(102, N'Italy ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(103, N'Jamaica ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(104, N'Japan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(105, N'Jersey ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(106, N'Jordan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(107, N'Kazakhstan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(108, N'Kenya ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(109, N'Kiribati ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(110, N'Korea, North ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(111, N'Korea, South ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(112, N'Kuwait ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(113, N'Kyrgyzstan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(114, N'Laos ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(115, N'Latvia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(116, N'Lebanon ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(117, N'Lesotho ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(118, N'Liberia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(119, N'Libya ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(120, N'Liechtenstein ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(121, N'Lithuania ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(122, N'Luxembourg ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(123, N'Macau ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(124, N'Macedonia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(125, N'Madagascar ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(126, N'Malawi ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(127, N'Malaysia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(128, N'Maldives ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(129, N'Mali ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(130, N'Malta ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(131, N'Marshall Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(132, N'Martinique ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(133, N'Mauritania ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(134, N'Mauritius ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(135, N'Mayotte ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(136, N'Mexico ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(137, N'Micronesia, Fed. St. ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(138, N'Moldova ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(139, N'Monaco ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(140, N'Mongolia ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(141, N'Montserrat ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(142, N'Morocco ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(143, N'Mozambique ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(144, N'Namibia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(145, N'Nauru ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(146, N'Nepal ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(147, N'Netherlands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(148, N'Netherlands Antilles ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(149, N'New Caledonia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(150, N'New Zealand ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(151, N'Nicaragua ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(152, N'Niger ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(153, N'Nigeria ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(154, N'N. Mariana Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(155, N'Norway ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(156, N'Oman ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(157, N'Pakistan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(158, N'Palau ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(159, N'Panama ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(160, N'Papua New Guinea ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(161, N'Paraguay ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(162, N'Peru ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(163, N'Philippines ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(164, N'Poland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(165, N'Portugal ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(166, N'Puerto Rico ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(167, N'Qatar ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(168, N'Reunion ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(169, N'Romania ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(170, N'Russia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(171, N'Rwanda ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(172, N'Saint Helena ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(173, N'Saint Kitts & Nevis ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(174, N'Saint Lucia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(175, N'St Pierre & Miquelon ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(176, N'Saint Vincent and the Grenadines ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(177, N'Samoa ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(178, N'San Marino ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(179, N'Sao Tome & Principe ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(180, N'Saudi Arabia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(181, N'Senegal ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(182, N'Serbia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(183, N'Seychelles ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(184, N'Sierra Leone ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(185, N'Singapore ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(186, N'Slovakia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(187, N'Slovenia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(188, N'Solomon Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(189, N'Somalia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(190, N'South Africa ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(191, N'Spain ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(192, N'Sri Lanka ');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(193, N'Sudan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(194, N'Suriname ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(195, N'Swaziland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(196, N'Sweden ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(197, N'Switzerland ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(198, N'Syria ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(199, N'Taiwan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(200, N'Tajikistan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(201, N'Tanzania ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(202, N'Thailand ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(203, N'Togo ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(204, N'Tonga ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(205, N'Trinidad & Tobago ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(206, N'Tunisia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(207, N'Turkey ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(208, N'Turkmenistan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(209, N'Turks & Caicos Is ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(210, N'Tuvalu ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(211, N'Uganda ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(212, N'Ukraine ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(213, N'United Arab Emirates ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(214, N'United Kingdom ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(215, N'United States ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(216, N'Uruguay ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(217, N'Uzbekistan ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(218, N'Vanuatu ');
```

```
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(219, N'Venezuela ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(220, N'Vietnam ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(221, N'Virgin Islands ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(222, N'Wallis and Futuna ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(223, N'West Bank ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(224, N'Western Sahara ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(225, N'Yemen ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(226, N'Zambia ');
INSERT INTO docFlowDB.docFlowSc.Countries (countryID, countryName) VALUES
(227, N'Zimbabwe ');
```

Дані таблиці Departments:

```
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (1,
N'Marketing');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (2,
N'IT');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (3,
N'Public Relations');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (4,
N'Export');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (5,
N'Import');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (6,
N'Procurement');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (7,
N'Finance');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (8,
N'Accounting');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (9,
N'HR');
INSERT INTO docFlowDB.docFlowSc.Departments (depID, depName) VALUES (10,
N'Logistic');
```

Дані таблиці docStates:

```
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (1,
N'Preparing');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (2,
N'Approval');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (3,
N'Signing');
```

```
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (4,
N'Accepted');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (5,
N'Registration');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (6,
N'For execution');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (7,
N'Archive');
INSERT INTO docFlowDB.docFlowSc.docStates (docStateID, docState) VALUES (8,
N'Refused');
```

Дані таблиці DocTypes:

```
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (1,
N'Contracts');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (2,
N'Documentation of bylaws');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (3,
N'Non-disclosure agreement');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (4,
N'Employment agreement');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (5,
N'Business plan');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (6,
N'Financial documents');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (7,
N'Transactional documents');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (8,
N'Compliance and regulatory documents');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (9,
N'Business reports');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (10,
N'Minutes of business meetings');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (11,
N'Contractor agreements');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (12,
N'Business insurance');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (13,
N'Partnership agreement');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (14,
N'Company policy');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (15,
N'Trademark');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (16,
N'Franchise agreement');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (17,
N'Employee stock option agreement');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (18,
N'Board resolutions');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (19,
N'Online terms of use');
```

```
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (20,
N'Business pitch deck');
INSERT INTO docFlowDB.docFlowSc.DocTypes (docTypeID, docType) VALUES (21,
N'Business license');
```

Дані таблиці Documents:

```
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (1, 10000, N'2021-12-17', 17, null, 1, N'Amazing', N'Lorem ipsum dolor
sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut
labore et dolore magna aliqua. Quam quisque id diam vel quam elementum
pulvinar etiam. Fermentum iaculis eu non diam phasellus vestibulum. Sit amet
mattis vulputate enim nulla aliquet. Penatibus et magnis dis parturient.
Molestie at elementum eu facilisis sed. Nec ultrices dui sapien eget mi
proin. Enim blandit volutpat maecenas volutpat blandit. Quis auctor elit sed
vulputate mi sit amet mauris. Amet consectetur adipiscing elit duis tristique
sollicitudin nibh sit amet. Blandit libero volutpat sed cras.', 4);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (2, 10001, N'2021-12-17', 9, null, 2, N'Just read this', N'Lorem ipsum
dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Faucibus a pellentesque sit amet porttitor
eget dolor. Venenatis cras sed felis eget velit aliquet sagittis id
consectetur. Mollis nunc sed id semper risus in hendrerit. Eu non diam
phasellus vestibulum lorem sed risus ultricies. Ac placerat vestibulum lectus
mauris ultrices eros. Odio aenean sed adipiscing diam donec. Aliquam ut
porttitor leo a diam sollicitudin. Cum sociis natoque penatibus et magnis
dis. Morbi blandit cursus risus at ultrices mi tempus imperdiet. Leo urna
molestie at elementum eu facilisis. Donec ultrices tincidunt arcu non sodales
neque. Enim ut tellus elementum sagittis.', 7);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (3, 10002, N'2021-12-17', 2, null, 2, N'Need to review', N'Lorem ipsum
dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Pharetra magna ac placerat vestibulum
lectus mauris ultrices eros. In nisl nisi scelerisque eu ultrices vitae
auctor eu augue. Egestas pretium aenean pharetra magna ac placerat. Suscipit
tellus mauris a diam maecenas sed enim ut. Viverra vitae congue eu consequat
ac. A diam maecenas sed enim ut. Nibh praesent tristique magna sit amet
purus. Neque volutpat ac tincidunt vitae. Orci eu lobortis elementum nibh
tellus molestie nunc. Faucibus et molestie ac feugiat. Amet est placerat in
egestas erat imperdiet sed. Velit dignissim sodales ut eu sem integer vitae
justo.', 8);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (4, 10003, N'2021-12-17', 10, null, 4, N'Check it out', N'Lorem ipsum
dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Odio pellentesque diam volutpat commodo sed
egestas egestas. Felis bibendum ut tristique et egestas quis ipsum. Aliquet
lectus proin nibh nisl condimentum id venenatis a. Cras fermentum odio eu
```

```
feugiat pretium nibh. Congue mauris rhoncus aenean vel elit scelerisque
mauris. Tellus id interdum velit laoreet id donec ultrices. Et egestas quis
ipsum suspendisse ultrices gravida dictum fusce. Maecenas ultricies mi eget
mauris. Sagittis id consectetur purus ut faucibus pulvinar elementum integer.
Adipiscing commodo elit at imperdiet dui accumsan sit amet. Lorem sed risus
ultricies tristique nulla aliquet. At elementum eu facilisis sed odio morbi
quis commodo. Eros in cursus turpis massa. Id consectetur purus ut faucibus.
Lectus mauris ultrices eros in cursus turpis massa tincidunt. Consectetur
adipiscing elit duis tristique. A diam maecenas sed enim.', 2);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (5, 10004, N'2021-12-17', 16, null, 5, N'Man look at this', N'Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Faucibus in ornare quam viverra
orci. Dictumst vestibulum rhoncus est pellentesque elit ullamcorper dignissim
cras. Sed egestas egestas fringilla phasellus faucibus scelerisque eleifend
donec. Pellentesque diam volutpat commodo sed egestas egestas fringilla
phasellus faucibus. At quis risus sed vulputate odio ut. Elementum tempus
egestas sed sed risus pretium quam. Dignissim enim sit amet venenatis urna
cursus. Vulputate mi sit amet mauris commodo quis. Pellentesque habitant
morbi tristique senectus et netus et. Eu lobortis elementum nibh tellus
molestie nunc. Ligula ullamcorper malesuada proin libero nunc consequat. Eget
magna fermentum iaculis eu non diam phasellus. Massa enim nec dui nunc. Magna
ac placerat vestibulum lectus mauris ultrices eros. Diam sollicitudin tempor
id eu nisl nunc.', 1);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (6, 10005, N'2021-12-17', 2, null, 6, N'Look at this dude!', N'Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. Vitae elementum curabitur vitae
nunc sed. Risus quis varius quam quisque id diam vel. Aliquet lectus proin
nibh nisl condimentum id venenatis a condimentum. Massa vitae tortor
condimentum lacinia quis vel. Ullamcorper a lacus vestibulum sed arcu non
odio euismod. Adipiscing elit duis tristique sollicitudin nibh sit amet
commodo. Nibh sed pulvinar proin gravida. Viverra nibh cras pulvinar mattis
nunc sed blandit libero volutpat. Mauris rhoncus aenean vel elit scelerisque
mauris pellentesque. Egestas sed tempus urna et. Morbi blandit cursus risus
at ultrices mi.', 8);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (7, 10006, N'2021-12-17', 3, null, 7, N'Refuse this doc pls', N'Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. In iaculis nunc sed augue lacus
viverra vitae congue. Sociis natoque penatibus et magnis dis parturient. Quis
ipsum suspendisse ultrices gravida dictum. Eget nunc lobortis mattis aliquam
faucibus purus in massa tempor. Orci dapibus ultrices in iaculis nunc sed
augue lacus. Placerat orci nulla pellentesque dignissim enim sit amet
venenatis. In nulla posuere sollicitudin aliquam ultrices sagittis. Amet
cursus sit amet dictum sit amet justo donec enim. Vulputate enim nulla
aliquet porttitor lacus luctus. Leo duis ut diam quam nulla porttitor. At
erat pellentesque adipiscing commodo. Sit amet consectetur adipiscing elit
duis tristique. Nec sagittis aliquam malesuada bibendum arcu vitae. Magna ac
```

```
placerat vestibulum lectus mauris ultrices eros. Id donec ultrices tincidunt
arcu non sodales neque. Proin gravida hendrerit lectus a.', 2);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (8, 10007, N'2021-12-17', 6, null, 8, N'Just ignore it', N'Lorem ipsum
dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua. Tellus rutrum tellus pellentesque eu
tincidunt tortor. Vitae ultricies leo integer malesuada nunc vel risus
commodo viverra. Aliquet eget sit amet tellus cras adipiscing. Facilisis
gravida neque convallis a cras. Odio tempor orci dapibus ultrices in iaculis.
Nisl purus in mollis nunc sed id. Duis ut diam quam nulla porttitor massa id.
Et malesuada fames ac turpis egestas integer eget. Sit amet mauris commodo
quis imperdiet massa tincidunt nunc pulvinar. Mollis aliquam ut porttitor leo
a diam. Vitae ultricies leo integer malesuada.', 4);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (9, 10008, N'2021-12-17', 20, null, 9, N'Good idea, check it', N'Lorem
ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor
incididunt ut labore et dolore magna aliqua. A cras semper auctor neque vitae
tempus quam. Nulla facilisi cras fermentum odio eu feugiat pretium nibh
ipsum. Netus et malesuada fames ac turpis egestas sed tempus urna. Eu
scelerisque felis imperdiet proin fermentum leo vel orci. Feugiat vivamus at
augue eget arcu dictum. Sed enim ut sem viverra aliquet. Lorem dolor sed
viverra ipsum nunc aliquet bibendum enim. Suspendisse potenti nullam ac
tortor vitae purus. Turpis egestas maecenas pharetra convallis posuere morbi
leo. Ipsum dolor sit amet consectetur adipiscing. Odio tempor orci dapibus
ultrices in iaculis nunc sed. Dictum at tempor commodo ullamcorper a lacus
vestibulum sed arcu. Sed elementum tempus egestas sed sed.', 3);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (10, 10009, N'2021-12-17', 10, null, 10, N'Bad idea, refuse it',
N'Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod
tempor incididunt ut labore et dolore magna aliqua. Eu augue ut lectus arcu
bibendum at varius vel. Lectus quam id leo in vitae turpis massa. Nulla
pellentesque dignissim enim sit. Proin sed libero enim sed faucibus turpis in
eu mi. Lectus urna duis convallis convallis tellus id interdum. Sem nulla
pharetra diam sit amet. Id velit ut tortor pretium viverra suspendisse
potenti nullam ac. Faucibus turpis in eu mi bibendum. Turpis massa tincidunt
dui ut. Quam elementum pulvinar etiam non quam lacus suspendisse faucibus
interdum. Tempor id eu nisl nunc mi ipsum faucibus vitae aliquet. Cras
tincidunt lobortis feugiat vivamus at augue eget arcu dictum. Consectetur
adipiscing elit ut aliquam purus. Sollicitudin tempor id eu nisl nunc mi
ipsum. Donec adipiscing tristique risus nec feugiat in fermentum posuere
urna. Velit euismod in pellentesque massa. Ut sem nulla pharetra diam sit
amet nisl. Felis donec et odio pellentesque diam volutpat.', 4);
INSERT INTO docFlowDB.docFlowSc.Documents (docID, docRegNum, docDate,
docTypeID, docFile, docAuthorID, docTitle, docDescription, docCurStateID)
VALUES (11, 10014, N'2021-12-17', 3, null, 2, N'New Document For Today',
N'ssdsdads fedFUVSfyu', 1);
```

Дані таблиці DocWorkFlow:

```sql
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (1, 10, 4, N'2014-04-12', null, 6, 4);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (2, 1, 4, N'2007-02-03', N'2021-12-11', 6, 4);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (3, 4, 4, N'1983-02-26', null, 3, 2);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (4, 6, 4, N'1974-08-22', N'2021-12-11', 3, 8);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (5, 3, 4, N'2008-11-11', N'2021-12-12', 8, 8);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (6, 9, 4, N'2004-07-12', null, 3, 3);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (7, 10, 4, N'2021-12-11', N'2021-12-17', 4, 4);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (8, 3, 4, N'2021-12-12', N'2021-12-17', 8, 8);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (9, 4, 4, N'2021-12-11', null, 2, 2);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (10, 5, 4, N'1970-04-09', null, 1, 1);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (11, 6, 4, N'2021-12-11', N'2021-12-17', 8, 8);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (12, 7, 4, N'1994-09-09', null, 2, 2);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (13, 8, 4, N'1995-04-22', N'2021-12-17', 4, 4);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (14, 1, 4, N'2021-12-11', N'2021-12-17', 4, 4);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (15, 2, 4, N'2002-02-01', N'2021-12-17', 7, 7);
INSERT INTO docFlowDB.docFlowSc.DocWorkFlow (workflowID, workflowDocID,
workflowWorkerID, workflowDateStart, workflowDateEnd, workflowPrevStateID,
workflowNewStateID) VALUES (16, 11, 4, N'2021-12-17', null, null, 1);
```

Дані таблиці Ranks:

```sql
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (1, N'Chief Executive Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (2, N'Chief Operating Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (3, N'Chief Financial Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (4, N'Chief Technology Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (5, N'Chief Marketing Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (6, N'Chief Human Resources Officer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (7, N'Clerk');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (8, N'Accountant');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (9, N'HR');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (10, N'Junior Developer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (11, N'Middle Developer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (12, N'Senior Developer');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (13, N'Business Analytic');
INSERT INTO docFlowDB.docFlowSc.Ranks (rankID, rankName) VALUES (14, N'Product Manager');
```

Дані таблиці Workers:

```sql
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (1, N'Valeriy', N'Gusak', 23000, 1000, 41, 4, 9, N'2012-01-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (2, N'Viktor', N'Nosov', 11000, 200, 23, 11, 10, N'2005-05-17');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (3, N'Muhamed', N'Ali', 15000, 300, 32, 14, 2, N'1974-10-26');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (4, N'Anton', N'Chaban', 50000, 17650, 8, 1, 2, N'2019-11-10');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (5, N'Adam', N'Abraham', 25188, 216, 22, 3, 6, N'2021-12-11');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (6, N'Adrian', N'Allan', 22569, 64, 37, 2, 8, N'2021-12-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (7, N'Alan', N'Alsop', 5564, 723, 40, 7, 9, N'1970-02-20');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (8, N'Alexander', N'Anderson', 3666, 974, 10, 7, 1, N'1985-04-16');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (9, N'Andrew', N'Arnold', 21435, 676, 36, 12, 7, N'1987-05-25');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (10, N'Anthony', N'Avery', 13140, 797, 41, 10, 10, N'2019-02-23');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (11, N'Austin', N'Bailey', 28008, 481, 29, 3, 3, N'1983-04-18');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (12, N'Benjamin', N'Baker', 7467, 943, 32, 9, 1, N'1973-02-23');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (13, N'Blake', N'Ball', 23755, 767, 35, 6, 3, N'2021-01-05');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (14, N'Matthew', N'Ball', 16264, 354, 37, 9, 7, N'2001-08-08');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (15, N'Brandon', N'Berry', 27827, 653, 19, 4, 3, N'1985-05-13');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (16, N'Oleksandr', N'Horduz', 22729, 9999, 8, 14, 6, N'2021-12-12');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (17, N'Cameron', N'Blake', 27471, 300, 18, 5, 2, N'2008-10-18');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (18, N'Carl', N'Bond', 13271, 717, 31, 10, 2, N'1999-09-16');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (19, N'Charles', N'Bower', 3158, 665, 10, 8, 2, N'2021-12-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (20, N'Christian', N'Brown', 8517, 428, 27, 11, 8, N'1978-08-07');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (21, N'Christopher', N'Buckland', 8207, 328, 23, 11, 1, N'2020-06-08');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (22, N'Colin', N'Burgess', 7081, 743, 24, 8, 3, N'1979-06-28');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (23, N'Connor', N'Butler', 29000, 280, 27, 2, 5, N'1988-01-31');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (24, N'Dan', N'Cameron', 6412, 208, 12, 8, 6, N'2009-04-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (25, N'David', N'Campbell', 18513, 522, 19, 13, 4, N'2005-05-05');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (26, N'Dominic', N'Carr', 7877, 491, 33, 9, 7, N'2008-12-05');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (27, N'Dylan', N'Chapman', 19120, 700, 13, 12, 8, N'1970-10-29');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (28, N'Edward', N'Churchill', 4118, 536, 11, 7, 8, N'2017-06-17');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (29, N'Eric', N'Clark', 7289, 242, 19, 8, 8, N'1973-11-22');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (30, N'Evan', N'Clarkson', 16863, 451, 22, 8, 8, N'2021-12-11');
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname, workerSalary, workerBonuses, workerAddressID, workerCurrentRankID, workerCurrentDepID, workerDate) VALUES (40, N'Ivan', N'Bosker', 13333, 555, 15, 14, 4, N'2021-12-11');
```

```sql
INSERT INTO docFlowDB.docFlowSc.Workers (workerID, workerName, workerSurname,
workerSalary, workerBonuses, workerAddressID, workerCurrentRankID,
workerCurrentDepID, workerDate) VALUES (41, N'Josh', N'Bush', 2349, 666, 13,
4, 3, N'2021-12-12');
```