

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Західноукраїнський національний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Валянський Євген Тарасович

**Програмний засіб моделювання сховища даних для
управління кадрами підприємства / Data warehouse
modeling software for enterprise personnel management**

спеціальність: 123 – Комп'ютерна інженерія
освітньо-професійна програма – Комп'ютерна інженерія

Кваліфікаційна робота

Виконав: студент групи КІ-41
Є. Т. Валянський

Науковий керівник
К.т.н., старший викладач Савка Н.Я.

Кваліфікаційну роботу
Допущено до захисту
«__» _____ 20 __ р.

Завідувач кафедри
_____ О.М. Березький

ТЕРНОПІЛЬ - 2021

РЕЗЮМЕ

Кваліфікаційна робота на тему «Програмний засіб моделювання сховища даних для управління кадрами підприємства» зі спеціальності 123 «Комп'ютерна інженерія» освітнього ступеня «бакалавр» містить 97 сторінки пояснюючої записки, 33 рисунка, 23 таблиць, 3 додатки. Обсяг графічного матеріалу 2 аркуші формату А3.

Метою кваліфікаційної роботи є розробка проекту корпоративної комп'ютерної мережі підприємства на основі моделі виродженого ядра.

Методи дослідження включають методи фізичної і логічної структуризації комп'ютерних мереж, методи структурного програмування, теорія графів, елементи математичної логіки.

Практичне значення – результати роботи призначені для використання у відділі кадрів установи і повинна служити підвищенню ефективності роботи з інформацією про співробітників організації, а також для створення документів на основі інформації, занесеної в базу даних.

Розроблена система управління базами даних є ефективним засобом, що дозволяє значно спостити роботу відділу кадрів при веденні інформаційної системи управління даними.

Ключові слова: БАЗА ДАНИХ, СИСТЕМА УПРАВЛІННЯ БАЗАМИ ДАНИХ, ВІДДІЛ КАДРІВ, КОНЦЕПТУАЛЬНА МОДЕЛЬ, ЛОГІЧНА МОДЕЛЬ, ФІЗИЧНА МОДЕЛЬ, ДІАГРАМА ВІДНОШЕНЬ.

RESUME

Qualification work on "Software Application Modeling Data Warehouse for Enterprise Human Resources Management" on the specialty 123 "Computer Engineering" educational level "bachelor" contains 97 pages of explanatory note, 33 pictures, 23 tables, 3 addenda. The volume of graphical material 2 arcs A3 format.

The purpose of the qualification work is to develop a draft of the corporate computer network of the enterprise on the basis of a model of the generated core.

Research methods include methods of physical and logical structurization of computer networks, methods of structural programming, graph theory, elements of mathematical logic.

Practical value - the results of the work is designed for use in the human resources department of the organization and should serve to improve the efficiency of working with information about co-workers of the organization, as well as for the creation of documents based on information entered in the database.

The developed database management system is an effective tool that allows you to significantly improve the work of the HR department in the conduct of information system of data management.

Keywords: DATABASE, DATABASE MANAGEMENT SYSTEM, HR DADLE, CONCEPTUAL MODEL, LOGICAL MODEL, PHYSICAL MODEL, DIAGRAM OF CONNECTIONS.

ЗМІСТ

Перелік умовних скорочень	8
Вступ.....	9
1 Сховища даних для управління кадрами на підприємстві та засоби їх моделювання.....	12
1.1 Особливості функціонування відділу кадрів на підприємстві.....	12
1.2 Аналіз засобів моделювання баз даних управління кадрами.....	15
1.3 Реляційні бази даних для проектування сховищ даних	21
2 Архітектура сховища даних для управління кадрами підприємства.....	26
2.1 Концептуальна модель	26
2.2 Логічна модель	29
2.3 Фізична модель.....	36
3 Програмна реалізація архітектури сховища даних для управління кадрами	46
3.1 Структура програмного модуля	46
3.2 Проектування сховища даних відділу кадрів на основі SQL	50
3.3 Управління відділом кадрів на підприємстві на основі розробленого сховища даних	56
4. Техніко-економічне обґрунтування розробки програмного засобу	62
4.1 Розрахунок витрат на розробку програмного забезпечення	62
4.2 Розрахунок експлуатаційних витрат і ціни	68
4.3 Визначення показників економічної ефективності	72

					КР.КІ. 07091/17.00.00.000 ПЗ			
Змн.	Лист	№ докум.	Підпис	Дата				
Розробив	Валянський				ПРОГРАМНИЙ ЗАСІБ МОДЕЛЮВАННЯ СХОВИЩА ДАНИХ ДЛЯ УПРАВЛІННЯ КАДРАМИ ПІДПРИЄМСТВА	Літ.	Арк.	Акрушів
Перевір.	Савка Н.Я.						6	95
Консульт.	Савка Н.Я.					ЗУНУ.ФКІТ. КІ-41		
Н. Контр.	Мельник Г.М.							
Затвердив	Березький О.М.							

Висновки	74
Список використаних джерел	76
Додаток А. ER – модель відношень відділу кадрів	80
Додаток Б UML – діаграма варіантів використання відділу кадрів	81
Додаток В Світлокопія виданих публікацій	82
Додаток Г Лістинг коду створення реляційної БД	83

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		7

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	–	База даних
СУБД	–	Система управління базами даних
ER	–	Entity-Relationship model
SQL	–	Structured Query Language
NoSQL	–	Бази даних, які не використовують мову запитів
IT	–	Information Technology
UML	–	Unified Modeling Language
ЗП	–	Заробітна плата
ПЗ	–	Погромне забезпечення

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Сучасні тенденції розвитку суспільства визначають інформацію, як основоположний елемент будь-якої діяльності. Наразі інформація є найкориснішим та найперспективнішим ресурсом людського буття. Існує багато видів інформації, традиційно, визначають вид в залежності від виду діяльності людини, яка з нею працює. Так, існує технічна інформація, яка використовується у комп'ютерній галузі, виробнича та управлінська, якою користуються на великих підприємствах та в установах, соціальна, правова тощо.

Однією із важливих видів інформації на підприємстві є інформація про кількісний та якісний склад працівників. Ефективна кадрова політика впливає на якість та результативність роботи всього підприємства та є довготривалою стратегією розвитку. Кадрова політика підприємства представляє собою набір системи принципів, моделей та цілей, що є визначальними при наборі та подальшій роботі з кадрами. Дана політика встановлюється власником підприємства, кадровою службою або вищим керівництвом. Основною метою кадрової роботи є своєчасне забезпечення підприємства персоналом потрібною якістю та чисельністю. З розвитком підприємства зростає чисельний склад робітників, що в свою чергу потребує подальшого вдосконалення інформаційної системи підприємств, які відповідають за облік кадрів.

Інформаційна система обліку кадрів містить такі складові:

- бази даних – загальний комплекс взаємопов'язаних даних про господарську діяльність та кадровий склад працівників;
- системи управління баз даних – спеціальну програму, що дозволяє накопичувати, структурувати, зберігати, опрацьовувати та надавати доступ до даних підприємства;

Комп'ютерна система обліку кадрів покликана реєструвати та обробляти

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		9

дані про кадровий склад підприємства. Накопичення інформації відбувається у базах даних, які впорядковують і організовують інформацію.

База даних (БД) – це структура призначення для зберігання і обробки інформації, великих і малих розмірів. Перевагою БД є висока швидкість доступу до інформації яка нам потрібна. Спеціальні алгоритми які використовуються для баз даних, дають змогу легко знаходити необхідні дані всього за лічені секунди[4].

Також в базі даних існує взаємозв'язок між інформацією: зміна в одному рядку автоматично змінює інший рядок – це допомагає працювати з даними швидше і простіше. Управління кадрами на підприємстві є одним із важливих засобів керування працівниками, включаючи вибір поставленої та звітної інформації а саме [12]:

- чітке розуміння й реалізація стратегічних цілей своєї фірми. Це дозволить реалізувати нові технології в присутній кадровій системі за короткий час;

- прогнозувати ситуації ринку праці. Без вивчення затрат праці та пропозиції висококваліфікованих робітників бажаного профілю, змін у мотивації праці та інших факторів просування трудових ресурсів можна втратити вже наявний кадровий потенціал;

- аналіз наявного кадрового потенціалу, планування його розвитку з урахуванням поглядів на майбутнє. Передусім, це планування природного руху кадрів – у зв'язку з навчанням, звільнення через хворобу, виходу на пенсію, службою у війську і т.д.

Це все і є підставою автоматизації підсистеми управління кадрами. Для підвищення швидкості і вдосконалення роботи керівника відділу кадрів створюються СУБД, в тому числі в програмах на основі SQL [14].

Мета кваліфікаційної роботи – обґрунтування та розробка інформаційного ресурсу для застосування на підприємстві, що стане інструментом для управлінців у процесі формування кадрового складу.

Відповідно до мети було визначено такі завдання:

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		10

- особливості функціонування відділу кадрів на підприємстві;
- аналіз засобів моделювання баз даних управління кадрами;
- можливості реляційних баз даних для проектування сховищ даних;
- проектування концептуальної, логічної та фізичної моделі для управління кадрами;
- реалізувати архітектуру сховищ даних для управління кадрами;
- управління відділом кадрів на підприємстві на основі розробленого сховища даних;
- техніко-економічне обґрунтування розробки програмного засобу.

Об'єкт дослідження – інформаційний ресурс в галузі організації кадрів підприємства.

Предмет дослідження – алгоритми та засоби моделювання сховищ даних для управління кадрами.

Методи дослідження – для вирішення поставлених завдань було використано такі методи: основні положення організації баз даних, методи порівняння та узагальнення, методи системного аналізу, методи синтезу та комп'ютерного моделювання.

Практичне значення – результати роботи призначені для використання у відділі кадрів установи і повинна служити підвищенню ефективності роботи з інформацією про співробітників організації, а також для створення документів на основі інформації, занесеної в базу даних.

За результатами роботи опубліковано тези доповіді на VIII Всеукраїнській науково-практичній конференції молодих науковців "ІТ-2021" [12]. Копії публікації наведено у додатку В.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		11

1 СХОВИЩА ДАНИХ ДЛЯ УПРАВЛІННЯ КАДРАМИ НА ПІДПРИЄМСТВІ ТА ЗАСОБИ ЇХ МОДЕЛЮВАННЯ

1.1 Особливості функціонування відділу кадрів на підприємстві

У підприємницькій діяльності використовують різні типи ресурсів: інтелектуальні, фінансові, інформаційні, матеріальні й найцінніший ресурс це люди, які роблять найважливіші завдання по управлінні всіма процесами. Відділ персоналу є ключовим у будь-якій компанії від залучення та утримання талантів до гарантування інтересів та добробуту працівників.

Щоб зрозуміти функції відділу кадрів, важливо знати, чого від нього очікувати та які цілі він має. Взагалі кажучи, ми знаємо, що головним його завданням є сприяння успіху компанії шляхом пошуку, підтримки та розвитку людського капіталу. З одного боку, це досягається шляхом створення ефективних програм, а з іншого, забезпечення відповідності встановленим правилам та процедурам у цій галузі [10].

Більш конкретними, найважливішими завданнями відділу кадрів є:

- підготовка документів та необхідних довідок про переведення працівників, підготовка та розробка необхідних бланків для цього;
- облік кадрів установи та її підрозділів;
- реєстрація та облік вхідних документів, що стосуються персоналу, нагляд за їх виконанням, в першу чергу виконання наказів та вказівок щодо персоналу;
- координація документообігу, забезпечення безпомилкового проходження документів і доручень керівництва по напряму роботи з кадрами та дотримання специфічних правил для напрямів роботи з кадрами;
- укладання номенклатурних операцій з кадрового діловодства, та їх оформлення;
- організація документів з кадрів для передачі в систему на зберігання;
- автоматизація і комп'ютерна обробка даних з особового складу.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		12

Перелік завдань практично нескінченний, тому можна сказати, що функції управління персоналом є численними та надзвичайно різноманітними. Всі ці функції повинні бути чіткими під час створення відділу кадрів та розробки стратегічного плану.

Адміністративне управління персоналом – одне з центральних завдань у відділі кадрів. Це включає управління контрактами, оплатою праці, дозволами, лікарняними, щорічними відпустками тощо [10].

Управління персоналом здійснює група фахівців, що виконують відповідну функцію, в якості працівників кадрової служби, а також керівників всіх підрозділів, що виконують роль керуючих по відношенню до своїх підлеглих.

Це суто адміністративна робота, споживає і вимагає належного рівня координації роботи з усією командою, працівниками та керівництвом, щоб уникнути помилок та проблем у майбутньому.

Велика кількість відділів людських ресурсів користуються електронними таблиці Excel для виконання цих завдань. Однак сьогодні програмне забезпечення управління людськими ресурсами дозволяє автоматизувати більшу частину цієї роботи та заощадити час і гроші приблизно на 30%.

Людські ресурси також відповідають за заробітну плату та виплати працівникам. Сюди входить розрахунок відповідних ставок оплати праці для кожної посади з урахуванням вартості, яку вони приносять компанії, зарплати, пропонованої на ринку праці, та параметрів, встановлених колективним договором [15].

Метою відділу кадрів є досягнення цілей організації шляхом забезпечення її необхідними працівниками та ефективного використання їх досвіду, майстерності, кваліфікації та творчого потенціалу.

Важливими завданнями відділу кадрів є:

– проаналізувати принципи створення електронних посібників та сучасні платформи дистанційно навчання;

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- організація відбору персоналу в необхідному об'ємі та необхідної майстерності;
- розробка шляхів кар'єрного росту працівників;
- створення ефективної системи управління штатним персоналом;
- пошук нових технологій по управлінні персоналом та інше.

До функцій відділу кадрів відноситься прийняття нових і звільнення старих співробітників, складання різних наказів і табеля робочого дня, сортування та облік документації по руху кадрів, контроль правильного формулювання наказів і розпоряджень в відділі кадрів, отримання та введення додаткової інформації про робітників які на сьогоднішній день закріплені за ним (дані про пенсіонерів, інвалідів) тощо [15].

Приймаючи нового працівника на підприємство підписує з ним трудовий договір (контракт), в якому знаходиться різна інформація про працівника, наприклад: коли працівник був прийнятий на роботу, ким він працює, умови роботи, розмір зарплати, табельний номер співробітника, дані про кількість днів відпустки працівника, чи компанія оплачує проїзд, відпустку і т.д.

Після офіційного прийняття співробітника на роботу заводиться особиста картка. Особиста картка має містити ім'я, прізвище, по батькові, паспортні дані, дата народження, місце прописки, проживання, склад сім'ї і т.д.

Протягом роботи працівник ходить у відрядження, займається різними курсами підвищення кваліфікації, йде у відпустки, хворіє, та може звільнитися. На все перераховане заводяться спеціальні накази (документи), які містять різну інформацію.

Комп'ютеризація відділу кадрів є однією з основних умов управління всіма процесами в організації та підняття можливостей кадрової служби, підвищення продуктивності та оперативності працівників.

Склад технічного та програмного забезпечення залежить від певних умов організаційної діяльності підприємства, а саме від обсягів виробництва, кількості кадрів, організаційної структури управління документообігом,

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		14

потребам в оперативній обробці інформації, централізації роботи з документами тощо.

Автоматизація управління кадрами повинно здійснюватися на підставі впорядкованої системи документування управлінської діяльності, уніфікації та зменшення кількості документів, створених в організації.

Впроваджувати комп'ютерні системи необхідно на всіх етапах управління, починаючи з підготовки документів до оперативної обробки, пошуку та реєстрації, швидкого транспортування, закінчуючи контролем виконанням та надійним зберіганням документації.

1.2 Аналіз засобів моделювання баз даних управління кадрами

Нові технології автоматизації систем управління інформацією впроваджуються там, де потрібно виконувати однотипні завдання, що передбачають запрограмованих рішень, або завдання з високим обсягом розрахунків або суто механічні роботи для відділу кадрів.

Оснащення фахівців автоматизованим робочим місцем дозволяє підвищити продуктивність праці працівників, скоротити їх чисельність і при цьому повисить швидкість обробки інформації та її достовірність, що необхідно для ефективного планування та управління [18].

Істотним недоліком неавтоматизованих інформаційних систем є унікальність обробки, збору та використання інформації. До того ж окремі працівники та підрозділи підприємства в цілому не мають доступу до інформації.

Основними цілями автоматизації є надання всім працівникам підприємства інформації на той час та у формі та обсязі, які йому потрібні (в межах рівня доступу до цієї інформації). Для цього потрібна єдина інтегрована база даних підприємства.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		15

При проектуванні нової бази даних необхідно розглянути ряд існуючих програмних засобів, які широко застосовуються для виконання завдань та цілей поставлених для управління кадрами підприємства [30].

Моделювання даних розвивалось поряд із системою управління базами даних, і типи моделей ускладняється у міру зростання потреб підприємництва у сховищі даних. Ось декілька типів моделей [14]:

- ієрархічна модель даних;
- мережева модель даних;
- реляційна модель даних.

Ієрархічна модель сховищ даних упорядковує дані у структуру дерева, з єдиним коренем, до якого пов'язані всі інші дані. Ієрархія починається з кореневих даних і розширюється як дерево, додаючи дочірні вузли до батьківських вузлів. У цій моделі допоміжний вузол матиме лише один батьківський вузол. Ця модель ефективно описує багато реальних відносин, таких як показчик керівник відділу, працівники тощо (рисунок 1.1).



Рисунок 1.1 – Ієрархічна модель даних

В ієрархічній моделі дані організовані у деревоподібну структуру з відношенням один до багатьох між двома різними типами даних, наприклад, підприємство може мати багато відділів, багато начальників і, звичайно, багато працівників.

Мережева модель – це продовження ієрархічної моделі. У цій моделі дані упорядковані більше як графік і дозволяється мати більше одного батьківського вузла (рисунок 1.2). У цій моделі бази даних дані більш пов'язані, оскільки в цій моделі бази даних встановлюється більше зв'язків.

Крім того, оскільки дані більш пов'язані між собою, отже, доступ до даних також простіший і швидший. Ця модель бази даних була використана для зображення взаємозв'язків даних. Це була найбільш поширена модель бази даних до того, як була представлена реляційна модель [29].

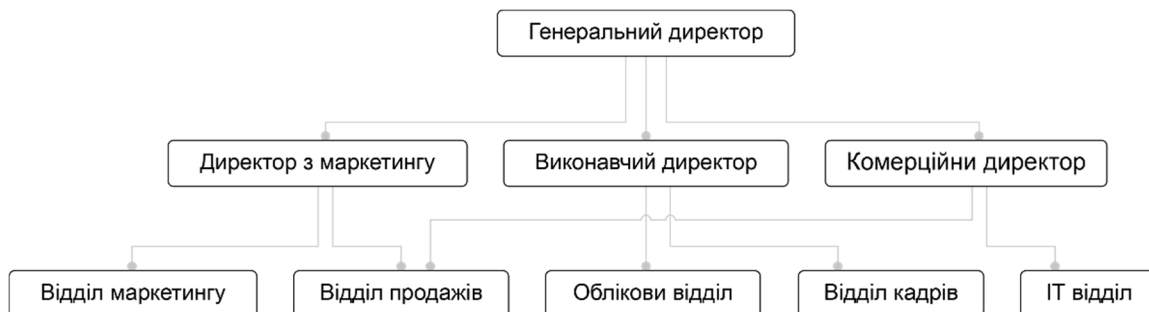


Рисунок 1.2 – Мережева модель даних

У реляційній моделі дані впорядковані у двовимірні таблиці, а взаємозв'язок підтримується шляхом зберігання загального поля. Ця модель була представлена EF Codd в 1970 році, і з того часу вона є найбільш широко використовуваною моделлю баз даних, насправді, можна сказати, єдиною моделлю бази даних, що використовується у всьому світі [18].

Основною структурою даних реляційної моделі є таблиці. Вся інформація, пов'язана з певним типом, зберігається в рядках цієї таблиці. Отже, таблиці також відомі як відношення в реляційній моделі.

Як і будь-який процес проектування баз даних та інформаційної системи починається з високого рівня абстракції й стає все більш конкретним та конкретним. Моделювання даних зазвичай можна розділити на три категорії, які видозмінюються залежно від абстракції [8].

Процес починається з концептуальної моделі й переходить до логічної моделі та завершиться фізичною моделлю. Кожен тип моделі даних обговорюється більш докладно нижче.

Концептуальні моделі даних. Вона також називаються моделями доменів і пропонують загальну картину того, що система буде містити, як вона буде організована та які ділові правила задіяні. Концептуальні моделі зазвичай

створюються як частина процесу збору початкових вимог проекту. Як правило, вони включають суттєві класи (визначають типи речей, які важливо представити в моделі даних), їх характеристики та обмеження, взаємозв'язки між ними та відповідні вимоги до безпеки та цілісності даних приклад на (рисунок – 1.3) [8].

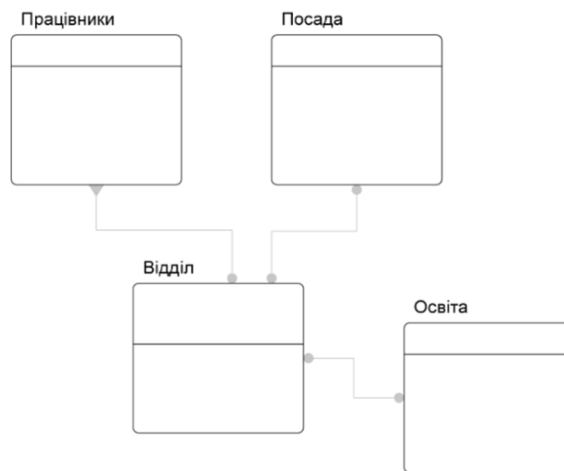


Рисунок 1.3 – Концептуальна модель даних

Логічна модель даних. Вона менш абстрактна та надає більш детальну інформацію про взаємозв'язки. Дотримується одна з декількох систем для позначення моделювання даних. Вони вказують атрибути даних, такі як типи даних та їх відповідні довжини, і показують зв'язки між сутностями дивитись (рисунок 1.4). Логічна модель даних не визначають жодних технічних вимог. Логічні моделі даних можуть бути корисними у високо процедурних середовищах які орієнтовані на дані за своєю природою, таких як проектування сховища даних або розробка систем звітності [8].

Фізична модель сховищ даних. Вона забезпечує схему фізичного збереження даних у базі даних. Таким чином, вона найменш абстрактна з усіх. Вона пропонує завершений дизайн, який може бути реалізований в реляційній базі даних, включаючи асоціативні таблиці, що показують взаємозв'язки між сутностями, а також первинними ключами та зовнішніми які будуть використовуватися для підтримки цих зв'язків приклад (рисунок 1.5). Фізичні

моделі даних можуть включати специфічні властивості системи управління базами даних (СУБД), включаючи продуктивність [8].

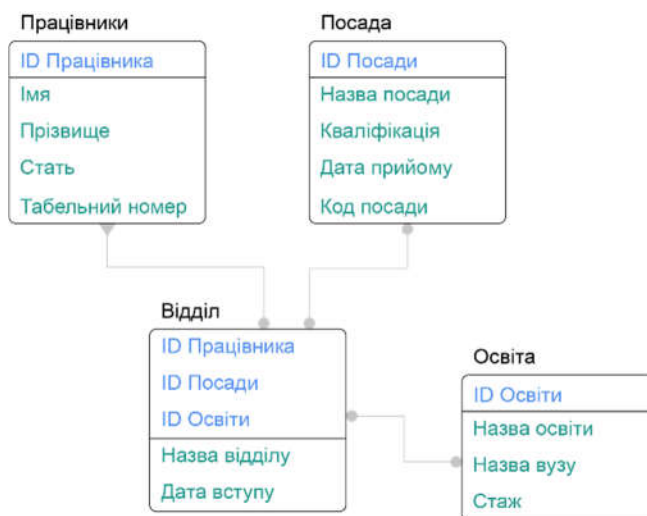


Рисунок 1.4 – Логічна модель даних



Рисунок 1.5 – Фізична модель сховищ даних.

Сьогодні широко використовуються численні комерційні та відкриті джерела програм для автоматизованої роботи над даними включаючи безліч інструментів моделювання та візуалізації. Ось декілька прикладів які представлені на ринку.

Microsoft SQL Server – цю систему керування базами даних, розробила компанія Microsoft. Її активно використовують в корпоративному секторі, особливо у великих компаніях. Ця СУБД – це цілий комплекс програм, що

дозволяє не тільки модифікувати й зберігати дані, але й аналізувати ці дані, здійснювати безпеку інформації та багато іншого [27].

MySQL – безкоштовна реляційна СУБД. Розробкою та підтримкою сайту MySQL здійснює компанія Oracle. MySQL широке поширення набула в інтернеті, як система зберігання даних у сайтів, іншими словами, переважна більшість сайтів зберігають свої дані в базі MySQL. Тому не дивно, що MySQL займає передову позицію нашого рейтингу [28].

Oracle Database – це система управління базами даних від компанії Oracle. Ця СУБД також активно використовується великими компаніями й стоїть немаленьких грошей, але натомість вона надає величезний функціонал і надійність. Тому Microsoft SQL Server і Oracle Database є серйозними конкурентами один одному [26].

PostgreSQL – це безкоштовна реляційна СУБД. Її активно використовують на UNIX-подібних платформах, для реалізації як малих і середніх, так і великих проектів [25].

MongoDB – це документ орієнтованих систем управління базами даних з відкритим вихідним кодом, яка не потребує опису схеми таблиць. MongoDB класифікується як NoSQL і використовує JSON-подібні документи та схему бази даних [1].

Redis – це сховище ключ-значення з відкритим кодом, яке часто використовується як шар хешування для роботи з іншим рішенням для зберігання даних. Основна причина, чому розробники роблять вибір на користь Redis, – це швидкість, яка набагато перевершує інші системи управління базами даних. Це також легко налаштувати, налаштувати та використовувати.

Вибір системи управління сховищами даних являє собою важкий вибір. Обрана система повинна задовольняти як наявним, так і майбутні потреби підприємства. Також слід враховувати фінансові витрати на придбання необхідного обладнання, розробку програмного забезпечення, а також навчання персоналу [3].

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		20

У результаті проведеного аналізу існуючих систем, виявлено недоліки й переваги, які будуть враховані при розробці нашої системи управління персоналом. При виборі систем управління персоналом необхідно враховувати: зручність для користувача, малу вартість, високу продуктивність, можливість захисту даних, з доступом у мережу.

1.3 Реляційні бази даних для проектування сховищ даних

Реляційна система управління базами даних - це загальноприйнятий тип баз даних, що зберігає дані в таблицях. Більшість баз даних, які сьогодні використовуються підприємствами, є реляційними базами даних, на відміну від простого файлу або ієрархічної бази даних. Більшість сучасних ІТ-систем та додатків базуються на реляційній СУБД [3].

Реляційні бази даних мають сили для обробки безлічі даних та складних запитів. Таблиці є стандартним використанням для сучасних баз даних. Дані часто зберігаються в багатьох таблицях, які також називаються "відносинами". Ці таблиці розділені на рядки, які також називаються записами та стовпцями (полями). У базі даних може бути мільйони рядків. Стовпці складаються з одного конкретного типу даних, наприклад, ім'я чи рік народження.

У таблицях даних, що використовуються в реляційній базі даних, сховища інформації пов'язані з ними об'єктів. Кожен рядок містить запис з унікальним ідентифікатором – відомий як ключ – і кожен стовпець містить атрибути даних. Кожен запис присвоює значення кожній ознаці, завдяки чому зв'язки між точками даних легко ідентифікувати.

Стандартний користувальницький інтерфейс та прикладна програма (API) реляційної бази даних є структурованою мовою запитів (SQL) [7]. Інструкції SQL використовуються як для інтерактивних запитів інформації з реляційної бази даних, так і для збору даних для звітів. Необхідно

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		21

дотримуватися визначених правил цілісності даних, щоб забезпечити точність та доступність реляційної бази даних.

Реляційні бази даних корисні для всіх типів і розмірів організацій, де пов'язаними точками даних необхідно послідовно керувати та захищати їх. Наприклад, підприємства електронної комерції можуть використовувати реляційну модель для обробки зарплати та відстеження цих запасів. Реляційна база даних є найбільш широко прийнятою моделлю бази даних.

Створюючи реляційну базу даних, ми можемо визначити область можливих значень у стовпці даних та подальші обмеження, які можуть застосовуватися до цього значення даних. Наприклад, домен можливих працівників може дозволити до 10 можливих імен працівників, але обмеження в одній таблиці, дозволяють вказати лише три з цих імен працівників [1].

Два обмеження належать до цілісності даних та первинного та зовнішнього ключів:

- цілісність сутності гарантує, що первинний ключ у таблиці є унікальним і що значення не має значення Null;
- цілісність посилок вимагає, щоб кожне значення у стовпці зовнішнього ключа знаходилось у первинному ключі таблиці, з якої воно походить.

Крім того, реляційні бази даних мають фізичну незалежність даних. Це стосується здатності системи вносити зміни у внутрішню схему без зміни зовнішніх схем або прикладних програм. Зміни внутрішньої схеми можуть включати:

- використання нових запам'ятовуючих пристроїв;
- модифікація індексів;
- перехід від конкретного методу доступу до іншого;
- використання різних структур даних;
- використано різні структури зберігання файлові організації.

Процес створення сховищ даних, що входить до загального життєвого циклу застосування, що використовує БД реляційного типу, охоплює основні

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		22

етапи проектування сховищ даних, а саме: побудова інфологічної та дата логічної моделей.

Розглянемо реляційну систему для реалізації відділу кадрів MySQL – це одна з найпопулярніших СУБД на сьогоднішній день, яка поширюється вільно (з деякими обмеженнями). Ця серверна система здатна ефективно функціонувати у взаємодії з інтернет-сайтами та вебдодатками. При цьому вона проста в освоєнні, що лише збільшує її популярність [3].

Результатом величезної популярності MySQL стала поява в інтернеті безлічі посібників з освоєння системи, а також величезної кількості всіляких плагінів і розширень, що спрощують роботу з цією системою. Це в свою чергу лише додає системі ще більшої популярності.

Одним з таких плагінів є MySQL Workbench. Він надає візуальні інструменти для створення, виконання та оптимізації запитів SQL [1]. Редактор SQL забезпечує виділення синтаксису кольорів, автоматичне заповнення, повторне використання фрагментів SQL та історію виконання SQL[7]. Панель підключень до бази даних дозволяє розробникам легко керувати стандартними підключеннями до бази даних, включаючи MySQL Fabric. Браузер об'єктів забезпечує миттєвий доступ до схеми бази даних та об'єктів.

Попри відсутність деякого функціоналу, наявного в інших СУБД, MySQL володіє досить великим розмаїттям доступних інструментів для створення баз даних. Також це безплатне рішення з відкритим кодом з RDBMS, яким володіє та керує Oracle. Незважаючи на те, що це безкоштовно, MySQL має перевагу від частоті безпеки та оновлень функцій. Комерційні та підприємства можуть перейти на платну версію MySQL, щоб отримати додаткові функції та підтримку користувачів

Крім поширеності й універсальності СУБД MySQL володіє цілим комплексом серйозних переваг перед іншими системами. Також слід відзначити такі якості як:

– простота у використанні. MySQL досить легко інсталується, а

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

наявність безлічі плагінів і допоміжних додатків спрощує роботу з базами даних;

- великий функціонал. Система MySQL має практично всім необхідним інструментарієм, який може знадобитися в реалізації практично будь-якого проекту;

- система безпеки створена таким чином, що безліч вбудованих функцій безпеки в ній працюють за замовчуванням;

- масштабованість. Будучи досить універсальною СУБД, MySQL досить легко і може бути використана для роботи з малими, та з великими об'ємами даних;

Як і будь-який програмний продукт, система MySQL має певні недоліки у своєму функціоналі, що не дозволяє використовувати її для роботи з додатками, що мають деякі специфічні вимоги. До недоліків цієї СУБД відносяться [14]:

- не ефективний в обробці дуже великих баз даних;
- не має настільки хорошого інструмента для розробки та налагодження в порівнянні з платними базами даних;
- недостатня надійність деяких процесів по роботі з базами даних поступається деяким іншим СУБД;
- низька швидкість розробки відносно платних засобів.

1.4 Постановка задачі кваліфікаційної роботи

Завданням кваліфікаційної роботи є розробка автоматичної системи управління відділом кадрів для оптимізації та покращення роботи підприємства. Як видно з проведеного аналізу система повинна мати зручний та зрозумілий інтерфейс, який забезпечить продуктивну роботу користувача та спростить управління кадрами. Тому, в кваліфікаційній роботі основним

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		24

завданням є аналіз продемонстрованих методів виявлення змін в схемі бази даних та аналіз методів відновлення на реагування до початкової схеми бази даних відділу кадрів.

Таким чином, на меті розробки функціоналу для запропонованого програмного продукту – інформаційної системи є допомога користувачам в процесі роботи через її оптимізацію, підвищення швидкості роботи, без втрат якості, що в результаті призведе до підвищення рівня управління.

Для досягнення мети необхідно виконати такі задачі:

- змоделювати концептуальну модель;
- змоделювати логічну модель;
- спроектувати фізичну модель;
- розробити структуру програмного модуля;
- спроектувати сховища даних на основі SQL;
- управління відділом кадрів на основі розробленого сховища даних;
- техніко-економічне обґрунтування розробки програмного засобу.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		25

2 АРХІТЕКТУРА СХОВИЩА ДАНИХ ДЛЯ УПРАВЛІННЯ КАДРАМИ ПІДПРИЄМСТВА

2.1 Концептуальна модель

Перехід на інтелектуальні системи дозволяє виключити суб'єктивний підхід до визначення якості персоналу, постійний доступ керівництва до інформації про персонал дає змогу підвищити інтелектуальний аналіз відділу кадрів та формувати різноманітної документації, необхідної для функціонування підприємство.

При підрахунку показників кадрового забезпечення та підготовки кадрів необхідні знання структури, а також кількісні та якісні характеристики працівників.

Концептуальний дизайн – це створення концептуального подання бази даних, що включає визначення типів важливих сутностей та відносин, які існують між ними. Концептуальна модель застосовується для структурування предметної області з урахуванням інформаційних інтересів користувачів БД, вона не залежить ні від програмних, ні від технічних помилок.

Першим етапом створення концептуальної моделі є вибір сутностей предметної області. Проектована інформаційна система може містити такі інформаційні об'єкти (сутності, ІО) [11]:

Працівники – включає інформацію яка описує працівників. Ця сутність включає такі атрибути: стать, ім'я, по батькові, прізвище, іноземна мова, сімейний стан, професія, табельний номер, дата прийняття на роботу, дату звільнення, коментарі.

Паспортні дані – ці дані містять інформацію про особу і документи які його ідентифікують. Ця сутність включає такі атрибути: серія паспорта, номер паспорта, дата видачі паспорта, дата народження, орган-видавець паспорту, адреса реєстрації, громадянство.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		26

Контактні дані – дані про місце проживання і номери телефонів. Ця сутність включає такі атрибути: номер телефону, фактичне місце проживання, електронна пошта.

Трудовий договір – цей договір містить дані про укладений з працівником трудовий договір. Ця сутність включає такі атрибути: серія договору, номер договору, дата укладення, заробітна плата, дата звільнення, причина звільнення.

Сім'я – містить інформацію про склад сім'ї працівника (тільки про близьких родичів). Ця сутність включає такі атрибути: повне ім'я, дату народження, місце народження, який член сім'ї.

Освіта – містить інформацію про навчальні заклади в яких працівник навчався. Ця сутність включає такі атрибути: назва навчального закладу, рік початку навчання, рік завершення навчання, серія і номер диплому, спеціальність, кваліфікаційний рівень.

Лікарняні – інформація про кількість днів проведеною працівником на лікарняному. Ця сутність включає такі атрибути: дата початку, дата закінчення, назва захворювання.

Штатний розклад – інформація про розклад працівника. Ця сутність включає такі атрибути: час початку і закінчення робочого дня, кількість працівників.

Посада – інформація про яку посаду займає працівник. Ця сутність включає такі атрибути: посада, заробітна плата.

Відпустка – інформація про відпустку, проведеною працівником. Ця сутність включає такі атрибути: дата початку, дата закінчення.

Нагороди – інформація про премії й здобуті нагороди під час роботи. Ця сутність включає такі атрибути: назва нагороди, дата отримання, сума премії.

Відділ – інформація про відділ в якому працює працівник. Ця сутність включає такі атрибути: назва відділу, керівник відділу, номер телефона відділу.

Пільги – інформація про пільги працівника. Ця сутність включає такі атрибути: назва пільг, кількість пільг.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

Документи – містить інформацію про документи, ксерокопії паспорта індикаційного коду і т.д. Ця сутність включає такі атрибути: назва документа, серія документа, номер документа.

Другим етапом є визначення зв'язків між сутностями [11].

Працівник та його робота є частиною системи кадрового забезпечення, вона показує поточну роботу працівника та його повноваження, а також інформацію про його робочий процес (рисунк. 2.1).



Рисунок 2.1 – Сутності й зв'язки трудової діяльності працівника

Також немалу роль в роботі працівника займає його соціальне життя ця частина містить інформацію про контактні дані, сімейне життя його пільги й нагороди (рисунк. 2.2).



Рисунок 2.2 – Сутності й зв'язки соціального життя працівника

Працівник та його освіта, ця частина відображає інформацію про всю освіту, отриману працівником, а також проходження різних курсів та сертифікацій (рисунок. 2.3).

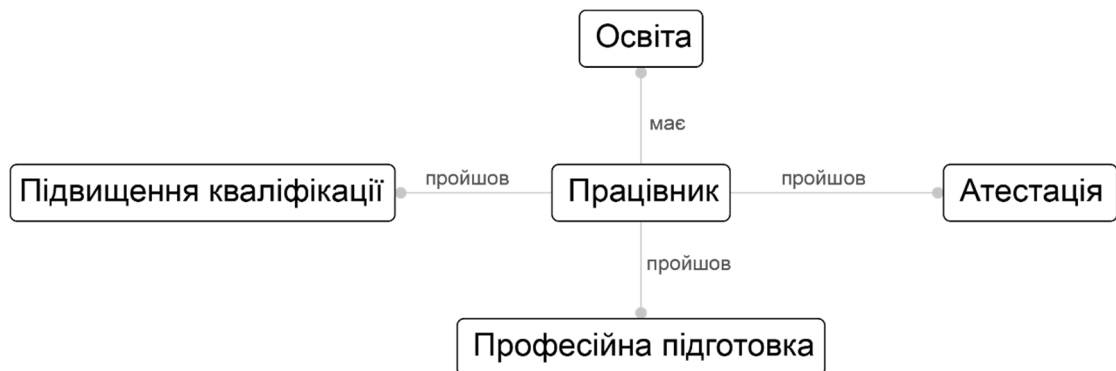


Рисунок 2.3 – Сутності й зв'язки підсистеми освіта працівника

Виявивши об'єкти, які представляють найбільший інтерес для даної бази даних, можна переходити до побудови моделі «сутність–зв'язок» [11]. Діаграма «сутність–зв'язок» є однією з ключових концепцій, які використовуються в процесі розробки бази даних. За допомогою цієї діаграми структура даних може бути чітко спроектована перед тим, як модель буде реалізована у вигляді бази даних. Діаграма «сутність–зв'язок» включає в себе логічні об'єкти, які були використані для представлення в базі даних об'єктів реального світу, і взаємозв'язки між цими об'єктами. У випадку проектування бази даних для обліку кадрів зазначена діаграма визначає взаємовідносини працівника.

2.2 Логічна модель

Логічне проектування бази даних полягає у перетворенні концептуальної моделі сховищ даних у логічну модель даних предметної

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		29

області з урахуванням обраного типу бази даних (обрано використання реляційної бази даних). У ньому також розглядаються ключові питання, пов'язані з побудовою системи ділових правил. Логічна модель даних є джерелом інформації для етапу фізичного проектування. Він надає розробнику фізичної моделі даних засоби для проведення всебічного аналізу різних аспектів роботи з даними, що дуже важливо для вибору ефективного проектного рішення.

Метою логічного моделювання є забезпечення найбільш природних способів збору та представлення інформацію, яка повинна зберігатися у створеній базі даних. Тому логічну модель даних намагаються побудувати за аналогією із природним поданням (останнє не можна використовувати в чистому вигляді через складність комп'ютерної обробки текстів та неоднозначність будь-якої природної мови).

Основними структурними елементами логічної моделі є:

- зв'язки між ними;
- сутність;
- їх властивості (атрибути).

Проект логічної структури бази даних встановлює склад реляційних таблиць, їх структуру та логічні зв'язки між таблицями. При формуванні структури кожної таблиці визначається набір полів (стовпців), для кожного з яких дається опис та тип, розміру даних та інших властивостей. Крім того, повинен бути вказаний унікальний ключ таблиці, який може складатися з одного або декількох полів.

Проектування проводиться на папері й вимагає дуже ретельного підходу, оскільки тут буде розроблена база даних. Для цього необхідно зібрати інформацію про її структуру і рух документів, визначити об'єкти системи й зв'язку між ними; виявити властивості та характеристики об'єктів, за якими створити поля, складові вихідні таблиці; для кожного об'єкта визначити первинні ключі (поля) і провести нормалізацію (розбиття) вихідних таблиць, встановити зв'язки.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		30

Потім необхідно провести етап створення бази даних на комп'ютері, а саме: скласти структури таблиць і ввести їх в комп'ютер; розробити запити для відбору даних, звіти для друку, створити екранні форми роботи з даними; спланувати порядок ведення і підтримання бази даних в робочому стані.

Таблиці в базі даних певним чином пов'язані між собою, завдяки чому їх називають реляційними базами даних. Така база даних складається з взаємопов'язаних реляційних таблиць.

Реляційні таблиці представляються двовимірним масивам і володіє наступними особливостями:

- кожна комірка таблиці містить один елемент даних;
- всі клітинки одного стовпця містять однаковий тип даних певної довжини;
- кожен стовпець має унікальне ім'я;
- кожен рядок таблиці зберігає відомості, що належить до одного об'єкта;
- порядок проходження рядків і стовпців може бути довільним;
- однакові рядки в таблиці відсутні.

Таблиці баз даних складаються з полів та стовпців і рядків – записів. Кожне поле таблиці містить однорідні дані, і кожен запис, відображає набір даних, що стосуються одного конкретного об'єкта. (наприклад, таблиця співробітники).

Для ідентифікації записів у таблиці має бути принаймні одне поле, яке називається первинним ключем. Це поле використовується для зв'язку між різними таблицями. Поле, значення якого однозначно ідентифікує відповідний запис, називається простим ключем. Так, у таблиці «Співробітники» ключовим полем є поле «Код працівника». Для взаємозв'язку таблиць ключ першої таблиці, який називається первинним, може бути введений в структуру другої таблиці, а ключ другої таблиці (зовнішній) може бути введений у першу таблиці.

Групування даних в таблиці може бути виконана різними способами, однак вона повинна відповідати вимогам нормалізації. Нормалізацією називають формальний апарат обмежень, застосовуваний при створенні таблиць, застосовуваний при створенні таблиць, що дозволяє усунути дублювання і суперечливість збережених даних і забезпечує ефективність їх обробки. Нормалізація таблиці на дві або більше, що мають кращі властивості підключення модифікації, змін або видалення даних. Кінцевою метою нормалізації є отримання такого проекту бази даних, в якому кожен факт фігурує лише в одному місці, тобто виключається надмірність інформації.

Створювана база даних повинна виконувати функції в інтересах автоматизації видачі даних про організацію. Вона повинна мати простий і наочний користувальницький інтерфейс, мати мінімальні системні вимоги.

База даних, що складається з декількох таблиць, повинна мати між таблицьні зв'язки, що забезпечують цілісність всіх даних бази та автоматизацію завдань обслуговування. Існують чотири типи зв'язків між таблицями бази даних:

Зв'язок один до одного (1 : 1). Пара таблиць несе взаємозв'язок "один до одного", коли один запис у першій таблиці пов'язаний лише з одним записом у другій таблиці, а один запис у другій таблиці пов'язаний лише з одним записом у першій таблиці. На (рисунку 2.4) показаний загальний приклад взаємозв'язку «один до одного».

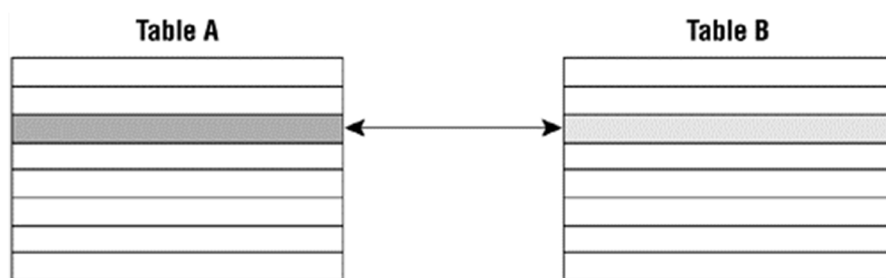


Рисунок 2.4 – Узагальнений приклад взаємин (1 : 1)

Як бачимо, один запис у таблиці А пов'язаний лише з одним записом у таблиці В, а один запис у таблиці В пов'язаний лише з одним записом у таблиці

А. Зв'язок «один до одного» зазвичай (але не завжди) включає таблицю підмножин.

Зв'язок один до багатьох (1 : N). Між парою таблиць існує взаємозв'язок "один до багатьох", коли один запис у першій таблиці може бути пов'язаний з одним або кількома записами в другій таблиці, але один запис у другій таблиці може бути пов'язаний лише з одним записом у першій таблиці. Розглянемо загальний приклад такого типу відносин.

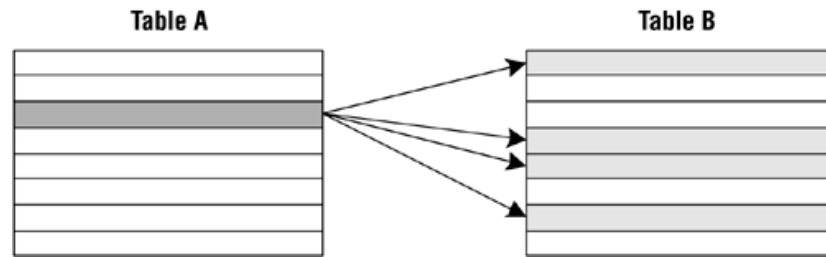


Рисунок 2.5 – Відношення (1 : N) з точки зору таблиці А

Скажімо працюємо з двома таблицями, таблицею А та таблицею В, які мають взаємозв'язок «один до багатьох». Через взаємозв'язок окремий запис у таблиці А може бути пов'язаний з одним або кількома записами в таблиці В. (Рисунок 2.5) показує зв'язок з точки зору таблиці А. І навпаки, один запис у таблиці В може бути пов'язаний лише з одним записом у таблиці А. На (рисунок 2.6) показано взаємозв'язок з точки зору таблиці В.

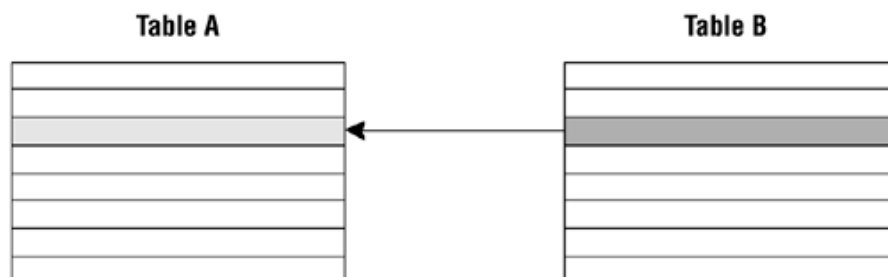


Рисунок 2.6 – Відношення (1:N) з точки зору таблиці В.

Зв'язок багато до багатьох (N:N). Пара таблиць несе відношення «багато до багатьох», коли один запис у першій таблиці можуть бути пов'язані з одним або декількома записами в другій таблиці і на оборот.

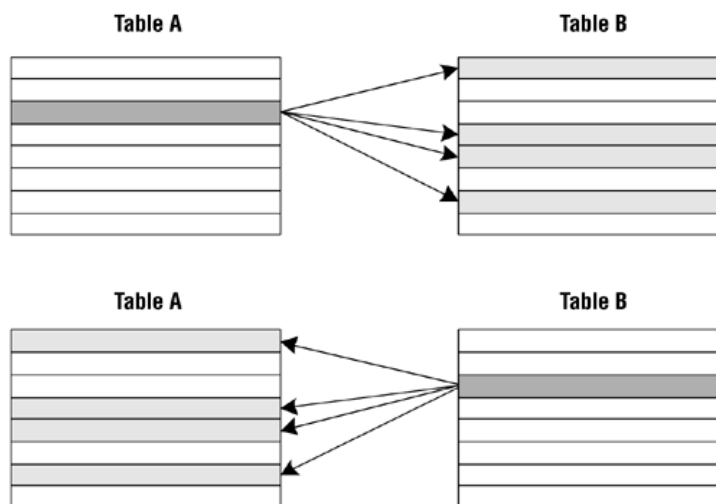


Рисунок 2.7 – Відносини (N : N) з точки зору таблиці А і таблиці В

Ще раз припустимо, що ви працюєте з таблицею А та таблицею В і що між ними існує взаємозв'язок «багато до багатьох». Через взаємозв'язок один запис у таблиці А може бути пов'язаний з одним або кількома записами (але не обов'язково всіма) в таблиці В. І навпаки, один запис у таблиці В може бути пов'язаний з одним або кількома записами (але не обов'язково всі) у таблиці А. На (рисунок 2.7) показано взаємозв'язок з точки зору кожної таблиці.

Якщо записи певні записи визначаються записами деяких полів, то така БД має комбінований ключ. Для зав'язків в реляційній таблиці ключ першого рядка таблиці називається первинним ключем він може вводитися в структуру другої таблиці, а ключ з другої таблиці (зовнішній ключ) може входити в першу таблицю.

Логічна модель даних може бути реляційної, ієрархічної або мережевою. Користувачам виділяються підмножини цієї логічної моделі, які називаються зовнішніми моделями (в деяких джерелах їх також називають підсхема), що відображають їхнє представлення про предметну область. Зовнішня модель відповідає уявленням, які користувачі отримують на основі логічної моделі, в той час, як концептуальні вимоги відбивають уявлення, які користувачі спочатку бажали мати і які лягли в основу розробки.

Моделювання предметної області базується на використанні графічних ER-діаграм (Entity-Relationship, це сутність-зв'язок), що включають невелику кількість різних компонентів.

На підставі викладеного вище, створимо концептуальну модель БД «Відділ кадрів» (рисунок 2.8).

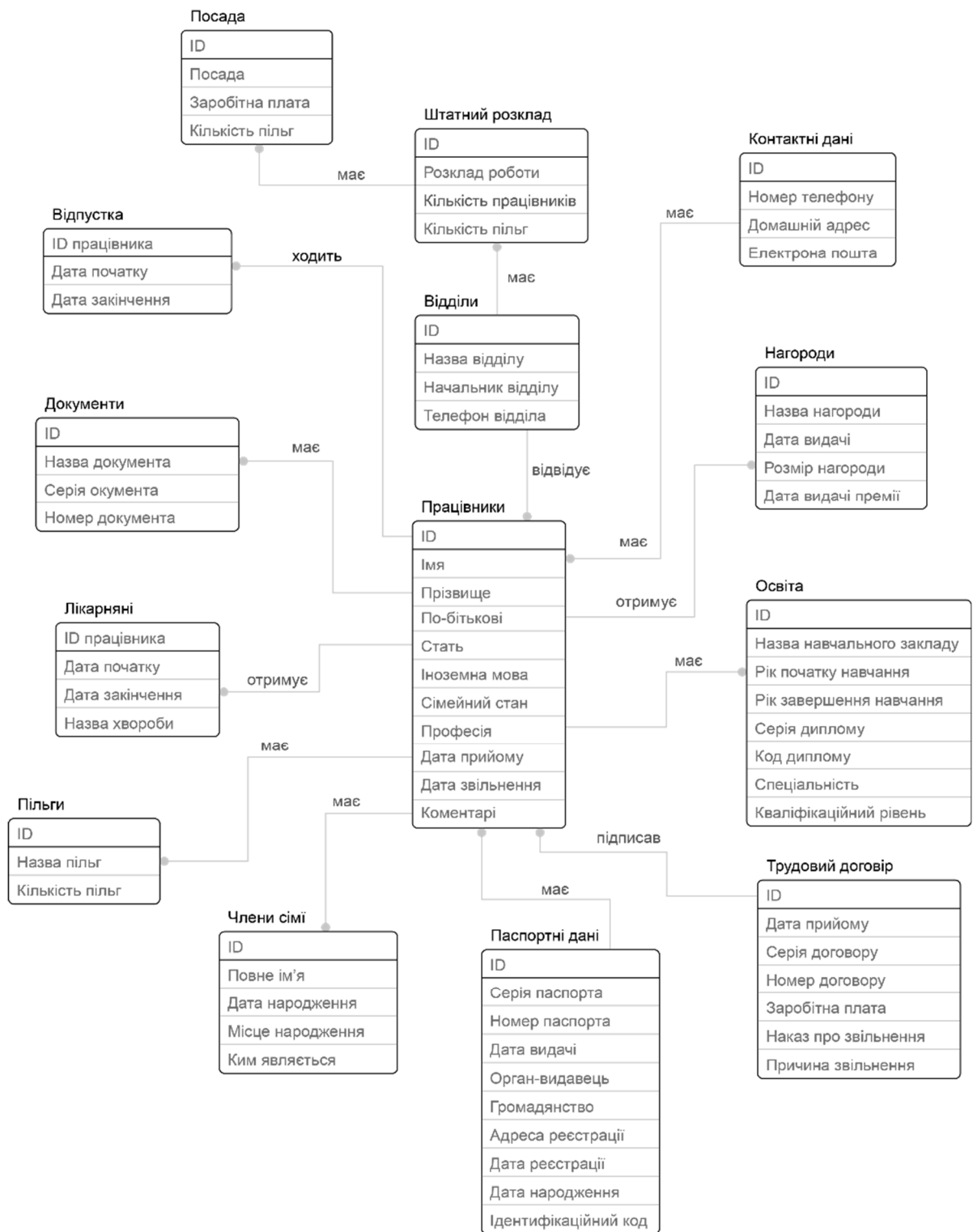


Рисунок 2.8 – Концептуальна модель відділу кадрів

Змн.	Арк.	№ докум.	Підпис	Дата

Після завершення етапу побудови логічної моделі бази даних можна переходити до побудови фізичної моделі. Логічне проектування рішення про те, як логічна модель буде реалізована у базі даних, що створюється з використанням вибраної предметної області.

2.3 Фізична модель

Фізичне проектування передбачає фактичне проектування бази даних відповідно до логічного моделювання. Логічне моделювання в основному передбачає збір інформації про підприємство, причому остання частина логічного моделювання спрямована на цілі та вимоги бази даних. Фізичне моделювання має справу з перетворенням логічної або бізнес-моделі в реляційну модель бази даних. Коли відбувається фізичне моделювання, об'єкти визначаються на рівні схеми. Схема – це група пов'язаних об'єктів у базі даних. Зусилля з проектування бази даних зазвичай пов'язані з однією схемою.

Під час фізичного моделювання такі об'єкти, як таблиці та стовпці, створюються на основі сутностей та атрибутів, які були визначені під час логічного моделювання. Також визначено обмеження, включаючи первинні ключі, зовнішні ключі, інші унікальні ключі та обмеження перевірки. Представлення можна створювати з таблиць бази даних, щоб узагальнити дані або просто надати користувачеві інший погляд на певні дані. Інші об'єкти, такі як індекси та знімки, також можуть бути визначені під час фізичного моделювання. Фізичне моделювання – це коли всі частини об'єднуються, щоб завершити процес визначення бази даних для підприємства.

Фізичне моделювання є специфічним програмним забезпеченням баз даних, що означає, що об'єкти, визначені під час фізичного моделювання, можуть змінюватися залежно від використовуваного програмного

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		36

забезпечення реляційних баз даних. Наприклад, більшість реляційних систем баз даних мають варіації способу представлення типів даних та способу зберігання даних, хоча основні типи даних концептуально однакові серед різних реалізацій. Крім того, деякі системи баз даних мають об'єкти, недоступні в інших системах баз даних.

Стадія фізичного проектування реалізацію обмежень і розробку основних відносин предметної області з застосуванням доступних функціональних засобів СУБД. На цій стадії ми повинні схвалити рішення щодо вибору методів отримання похідних даних, що містяться в моделі даних. Наступним кроком є вибір організації файлів та індексів для базових відносин. Як правило, системи управління базами даних для ПК мають фіксовану структуру зовнішньої пам'яті, а інші бази даних пропонують кілька альтернатив організації файлів для зберігання даних.

З точки зору користувача, організація внутрішньої структури зберігання відносин повинна бути повністю прозорою користувачі повинні мати доступ до будь-яких відносин та окремих рядків, не розраховуючи, як зберігати дані. Це означає, що системи управління повинні забезпечувати повну незалежність фізичного зберігання даних від їх логічної організації. Тільки в цьому випадку внесення змін до фізичної організації бази даних не вплине на роботу користувачів.

Відповідність між логічною моделлю даних та фізичною моделлю даних визначається внутрішньою схемою бази даних. Розробник надає докладні фізичні проекти бази даних з урахуванням використовуваної бази даних та операційної системи проекту реалізації бази даних в системі управління, розробник визначає файлові структури, які будуть використовуватися для представлення кожного відношення.

При реалізації проекту бази даних в операційній системі розробник вказує розташування окремих файлів і забезпечує необхідний захист. Наступним кроком є розробка захисних заходів, необхідних для запобігання несанкціонованому доступу до даних, включаючи контроль доступу до основних відносин. Необхідність зниження рівня вимог до нормалізації даних

у логічній моделі, що може підвищити загальну продуктивність системи. Однак ці кроки слід вживати лише за крайньої потреби, оскільки введення надлишкових даних до бази даних спричинить проблеми з підтримкою цілісності даних.

Приставаючи до фізичного проектування спочатку необхідно проаналізувати інформацію про зв'язки, зібрані на етапі побудови логічної моделі бази даних. Ця інформація може міститися у словнику даних та у визначеннях взаємозв'язків. У фізичній моделі дані відносин включають такі елементи [29]:

- список простих атрибутів;
- ім'я відносин;
- визначення первинного ключа(РК) і зовнішніх (FK) ключів;
- список похідних атрибутів та опис способів їхнього обрахунку;
- визначення вимог будь-яких зовнішніх ключів.

Для кожного атрибута в словнику даних повинна бути присутня наступна інформація:

- визначення його домену, що включає типу даних та розмірність;
- внутрішнє подання атрибута й припустимі обмеження;
- допустимість значення NULL для даного атрибута.

Перейдемо до проектування самої БД, а саме таблиць. База даних складається з наступних таблиць:

- 1) працівники, «employee»;
- 2) контактні дані, «contacts_data»;
- 3) паспортні дані, «passport_data»;
- 4) трудовий договір, «employment_contract»;
- 5) члени сім'ї, «family_members» ;
- 6) освіта, «education»;
- 7) лікарняні, «sick_leave»;
- 8) відділи, «department»;

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		38

9) штатний розклад, «staff_schedule»;

10) посада, «position»;

11) відпустка, «vacation»;

12) нагороди, «reward»;

13) пільги, «social_benefits»;

14) документи, «documents»;

Структура таблиці «employee» реалізуються в (таблиця 2.1) де зберігається інформація про працівників підприємства.

Таблиця 2.1 – Структура таблиці «employee»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Ім'я	name	VARCHAR(50)	Обов'язкове поле
Прізвище	surname	VARCHAR(50)	Обов'язкове поле
По батькові	patronymic	VARCHAR(50)	Обов'язкове поле
Стать	gender	VARCHAR(50)	Обов'язкове поле
Іноземна мова	foreign_language	VARCHAR(100)	Обов'язкове поле
Сімейний стан	reason_for_dismissal	VARCHAR(50)	Обов'язкове поле
Професія	profession	VARCHAR(200)	Обов'язкове поле
Дата прийняття	date_admission	DATE	Обов'язкове поле
Дата звільнення	marital_status	DATE	Не обов'язкове поле
Коментарі	comments	VARCHAR(500)	Не обов'язкове поле
Ідентифікатор департаменту	department_id	INT	Зовнішній ключ до department
Ідентифікатор паспорта	passport_data_id	INT	Зовнішній ключ до passport_data
Ідентифікатор контракт	contacts_data_id	INT	Зовнішній ключ до contacts_data

Структура таблиці «contacts_data» реалізуються в (таблиця 2.2) де зберігається інформація про контактні дані працівника.

Таблиця 2.2 – Структура таблиці «contacts_data»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Номер телефону	phone_number	VARCHAR(13)	Обов'язкове поле
Домашній адрес	home_address	VARCHAR(200)	Обов'язкове поле
Електронна пошта	email	VARCHAR(50)	Обов'язкове поле

Структура таблиці «passport_data» реалізуються в (таблиця 2.3) де зберігається інформація про паспортні дані працівника.

Таблиця 2.3 – Структура таблиці «passport_data»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Серія паспорта	passport_series	VARCHAR(2)	Обов'язкове поле
Номер паспорта	passport_number	DATE	Обов'язкове поле
Дата видачі	date_of_issue	DECIMAL(10,2)	Обов'язкове поле
Ким виданий	issued_by	VARCHAR(100)	Обов'язкове поле
Громадянство	citizenship	VARCHAR(50)	Обов'язкове поле
Адрес реєстрації	registration_address	VARCHAR(200)	Обов'язкове поле
Дата народження	date_of_birth	DATE	Обов'язкове поле

Структура таблиці «employment_contract» реалізуються в (таблиця 2.4) де зберігається інформація про контракт який він підписав з підприємством.

Таблиця 2.4 – Структура таблиці «employment_contract»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Дата прийняття	date_of_conclusion	DATE	Обов'язкове поле
Серія договору	contract_series	VARCHAR(2)	Обов'язкове поле
Номер договору	contract_number	INT	Обов'язкове поле
Заробітна плата	salary	DECIMAL(10,2)	Обов'язкове поле
Причина звільнення	reason_for_dismissal	VARCHAR(45)	Не обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «family_members» реалізуються в (таблиця 2.5) де зберігається інформація про сім'ю працівника.

Таблиця 2.5 – Структура таблиці «family_members»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Повне ім'я	full_name	VARCHAR(200)	Обов'язкове поле
Дата народження	date_of_birth	DATE	Обов'язкове поле
Місце народження	place_of_birth	VARCHAR(200)	Обов'язкове поле
Ким являється	who_is	VARCHAR(50)	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «education» реалізуються в (таблиця 2.6) де зберігається інформація про освіту.

Таблиця 2.6 – Структура таблиці «education»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Назви навального закладу	name_institution	VARCHAR(200)	Обов'язкове поле
Рік початку навчання	start_of_training	VARCHAR(10)	Обов'язкове поле
Рік закінчення навчання	completion_of_training	VARCHAR(10)	Обов'язкове поле
Серія диплому	diploma_series	VARCHAR(2)	Обов'язкове поле
Код диплому	diploma_code	INT	Обов'язкове поле
Спеціальність	specialty	VARCHAR(100)	Не обов'язкове поле
Кваліфікаційний рівень	qualification_level	INT	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «sick_leave» реалізуються в (таблиця 2.7) де зберігається інформація про лікарняні.

Таблиця 2.7 – Структура таблиці «sick_leave»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Дата початку	start_date	DATE	Обов'язкове поле
Дата закінчення	end_date	DATE	Обов'язкове поле
Назва захворювання	name_of_disease	VARCHAR(500)	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «department» реалізуються в (таблиця 2.8) де зберігається інформація про відділ в якому він працює.

Таблиця 2.8 – Структура таблиці «department»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор відділу	department_id	INT	Унікальний ключ
Назва відділу	name_department	VARCHAR(50)	Обов'язкове поле
Начальник відділу	head_of_department	VARCHAR(50)	Обов'язкове поле
Телефон відділу	telephone_department	VARCHAR(13)	Обов'язкове поле

Структура таблиці «staff_schedule» реалізуються в (таблиця 2.9) де зберігається інформація про штатний розклад.

Таблиця 2.9 – Структура таблиці «staff_schedule»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Розклад роботи	work_time	VARCHAR(15)	Обов'язкове поле
Кількість працівників	number_of_employees	INT	Обов'язкове поле
Ідентифікатор розкладу	department_id	INT	Зовнішній ключ до department

Структура таблиці «position» реалізуються в (таблиця 2.10) де зберігається інформація про посаду.

Таблиця 2.10 – Структура таблиці «position»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Посада	position	VARCHAR(45)	Обов'язкове поле
Заробітна плата	salary	DECIMAL(10,2)	Обов'язкове поле
Ідентифікатор розкладу	staff_schedule_id	INT	Зовнішній ключ до staff_schedule

Структура таблиці «vacation» реалізуються в (таблиця 2.11) де зберігається інформація про відпустки.

Таблиця 2.11 – Структура таблиці «vacation»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Дата початку	start_date	DATE	Обов'язкове поле
Дата закінчення	end_date	DATE	Не обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «reward» реалізуються в (таблиця 2.12) де зберігається інформація про нагороди.

Таблиця 2.12 – Структура таблиці «reward»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Назва нагороди	name_of_reward	VARCHAR(200)	Обов'язкове поле
Дата видачі	date_of_issue	DATE	Обов'язкове поле
Розмір нагороди	size_of_reward	DECIMAL(10,2)	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «social_benefits» реалізуються в (таблиця 2.13) де зберігається інформація про пільги.

Таблиця 2.13 – Структура таблиці «social_benefits»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Назва пільг	name_of_benefits	VARCHAR(200)	Обов'язкове поле
Кількість пільг	number_of_benefits	DECIMAL(2,1)	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Структура таблиці «documents» реалізуються в (таблиця 2.14) де зберігається інформація про документи працівника.

Таблиця 2.14 – Структура таблиці «documents»

Зміст поля	Ім'я поля	Тип, даних	Примітки
Ідентифікатор	id	INT	Унікальний ключ
Назва документа	name_document	VARCHAR(500)	Обов'язкове поле
Серія документа	document_series	VARCHAR(2)	Не обов'язкове поле
Номер документа	document_number	VARCHAR(20)	Обов'язкове поле
Ідентифікатор працівника	employee_id	INT	Зовнішній ключ до employee

Для побудови фізичної моделі використаємо програмний засіб MySQL Workbench, в результаті отримана схема показана на (рисунку 2.9).

Спроектвану базу даних для реалізації інформаційної системи можна вважати нормалізованою, оскільки її таблиці знаходяться в третій нормальній формі. Тобто неключові стовпці в таблиці не залежать від інших неключових стовпців, а залежать лише від первинного ключа. Отже, таблиці також у першій та другій нормальній формі.

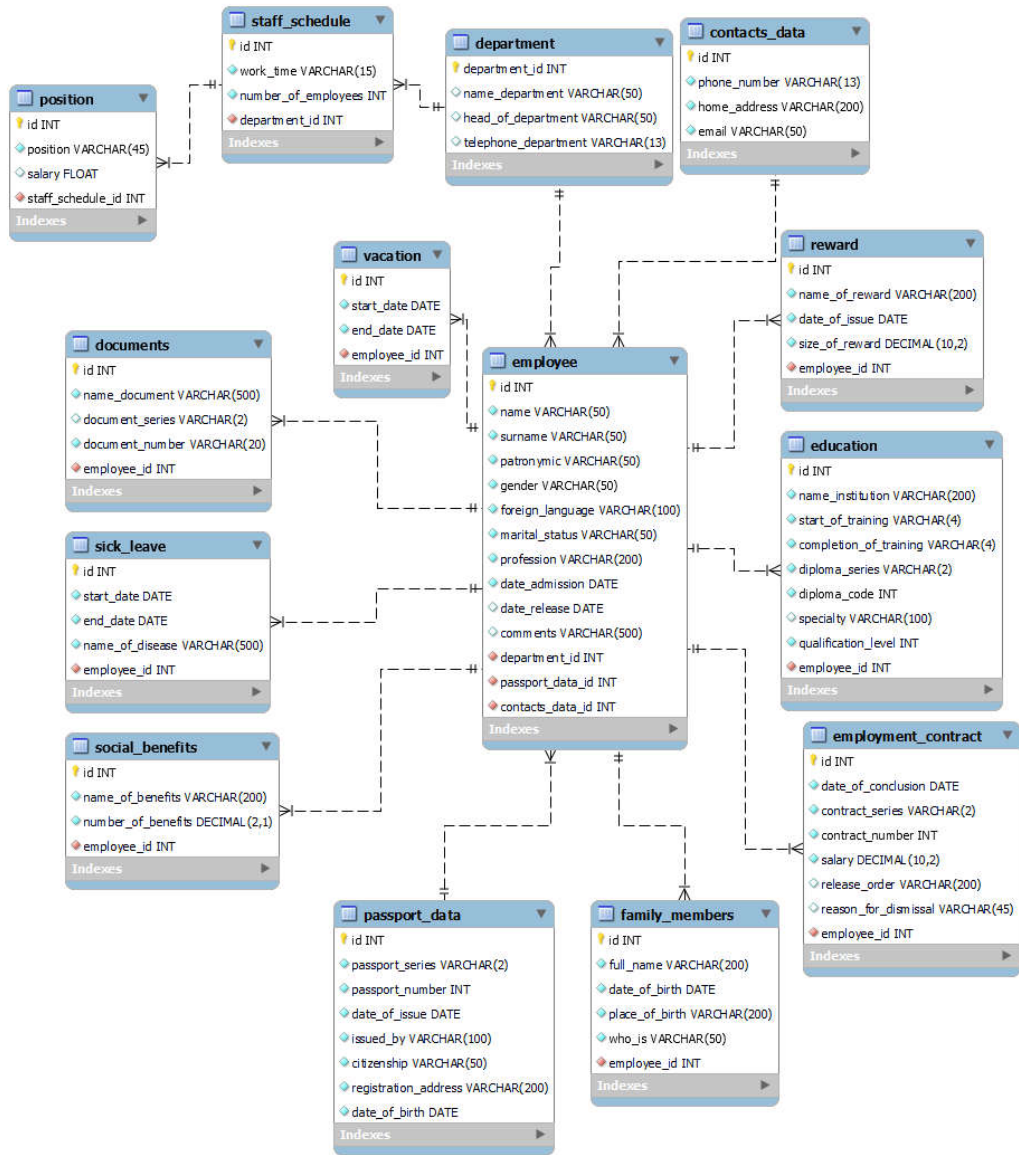


Рисунок 2.9 – Фізична модель у вигляді EER діаграми MySQL Workbench

Перша звичайна форма забороняє дублювання стовпців, забороняє кілька стовпців і вимагає вказати первинний ключ для таблиці. Друга звичайна форма вимагає, щоб не ключові стовпці таблиць висіли з первинного ключа в цілому, але не з його частини.

Змн.	Арк.	№ докум.	Підпис	Дата

3 ПРОГРАМНА РЕАЛІЗАЦІЯ АРХІТЕКТУРИ СХОВИЩА ДАНИХ ДЛЯ УПРАВЛІННЯ КАДРАМИ

3.1 Структура програмного модуля

Розробка будь-якої системи вимагає детального аналізу та розуміння не лише програмних засобів, але і предметної області в якій ведеться розробка. Проаналізувавши предметну область та усі функціональні, нефункціональні вимоги, можна приступати до моделювання різних етапів роботи системи за допомогою мови об'єктноорієнтованого моделювання UML.

UML – це об'єднана мова моделювання. Вона описує створення моделей, які описують об'єкти. Підходить для широкого класу розроблених програмних систем, різних сфер застосування, типів організацій, різних рівнів компетентності та розмірів проектів [10]. UML описує об'єкт в одному заданому синтаксисі, тому, де б ви не малювали діаграму, її правила будуть зрозумілі кожному, хто знайомий з цією графічною мовою – навіть в іншій країні.

Конструкції UML створюються з чисельних елементів моделі, що позначають різні частини програмної системи. UML використовуються для побудови діаграм, які відповідають певним частинам системи або точці зору на систему [10].

Для написання ключових варіантів використання для початку необхідно визначити акторів системи.

Актор – це суб'єкт у діаграмі UML який представляє ролі, коли він взаємодіє із системою та її об'єктами. Важливо пам'ятати що актор завжди знаходиться поза сферою системи, яку ми прагнемо моделювати за допомогою діаграми UML.

Актори використовуються, щоб зобразити різні ролі, включаючи користувачів та інші зовнішні взаємодії суб'єктів. Актори представляються в

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		46

діаграмі UML, використовуючи позначення особи з палицею. Схематичний вигляд UML актору зображено на (рисунку 3.1).

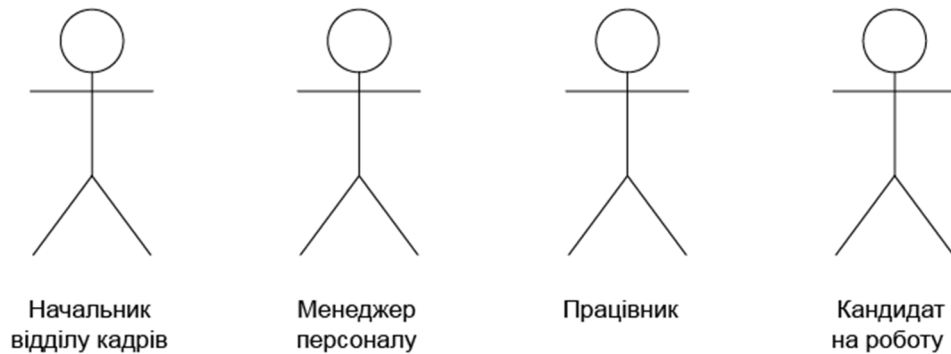


Рисунок 3.1 – Головні актори системи

Саме ці актори виступають головними користувачами відділу кадрів. В таблиці 3.1 наведено ключових акторів та описано їх головну роль в системі.

Таблиця 3.1 – Головні актори системи

Актор	Короткий опис
Начальник відділу кадрів	Користувач який може оформлювати прийняття, перевід, звільнення також він контролює роботи працівників і відділу кадрів.
Менеджер персоналу	Користувач який має змогу формувати записи про працівника заповнювати трудові договори оформлювати відпустки, лікарняні та приймати нових працівників.
Працівник	Користувач який може подати заяву про звільнення оформити лікарняні, відпустку, привести нового працівника.
Кандидат на роботу	Користувач який може подавати резюме і проходити співбесіди.

Діаграми прецедентів або іншими словами варіантів використання зазвичай називаються діаграмами поведінки, що використовуються для опису

набору дій (випадків використання), які деякі системи повинні або можуть виконуватись у співпраці з одним або декількома зовнішніми користувачами системи (суб'єктами) [10].

Кожен випадок використання повинен дати деякий помітний і цінний результат учасникам або іншим зацікавленим сторонам системи. Можна зауважити, що UML специфікації також описують варіанти використання як спеціалізації діаграми класів, а діаграма класів – це структурна схема. Також є структуровані діаграмами – як особливий випадок діаграм класів, де класифікатори мають бути або дійовими особами, або використовують випадки, пов'язані з асоціаціями. На основі визначених акторів в попередньому розділі можна побудувати ключові варіанти використання. Варіанти використання для відділу кадрів показані на (рисунку 3.2).

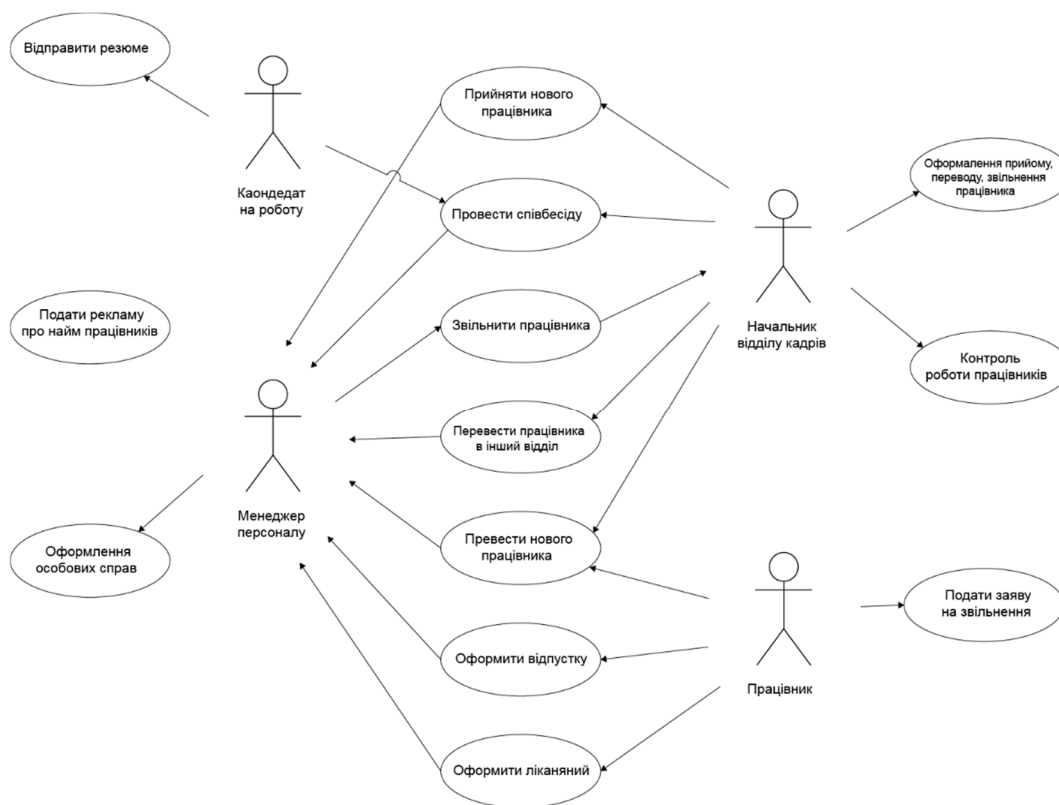


Рисунок 3.2 – Діаграма варіантів використання взаємодії відділу кадрів

На основі схеми діяльності відділу кадрів була розроблена схема послідовності. Вибрано такі суб'єкти:

– службовець;

- кандидат;
- менеджер з персоналу;
- звільнений;
- облаштування;
- неробочі дні.

Відділ кадрів займається підбором персоналу, набором, переведенням та звільненням працівників. Кандидат на вакансію надсилає резюме і приходить до менеджера з персоналу, який проводить з ним співбесіду. Якщо керівник визначив рівень кваліфікації кандидата і він підходить для компанії, менеджер повідомляє директора про кандидата, який приймає рішення. Потім менеджер проводить процедуру набору. Тепер працівника можна перевести, поїхати у відпустку, на лікарняний або звільнити [14].

Отже, якщо кандидат звернувся до компанії, він підписує контракт, інструкцію по телевізору, документ про відповідальність. Потім менеджер з персоналу реєструє нового співробітника в базі даних та оформляє особисту справу.

При необхідності менеджер з персоналу може перевести працівника на іншу посаду. Працівник подає заяву, і якщо є вакансія і працівник відповідає критеріям вакансії, працівник переводиться на іншу посаду.

Працівник має право на відпустку. Якщо йому потрібна відпустка, він пише заяву менеджеру з персоналу, який визначає термін відпустки. Потім працівник їде у відпустку.

Якщо працівник не може вийти на роботу через фізичний стан, він звернеться до лікарні. Після госпіталізації працівник видає довідку від лікаря.

У разі необхідності працівник може бути звільнений. Для звільнення працівника він не повинен виконувати свої обов'язки або написати заяву (рисунок 3.3).

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
						49
Змн.	Арк.	№ докум.	Підпис	Дата		

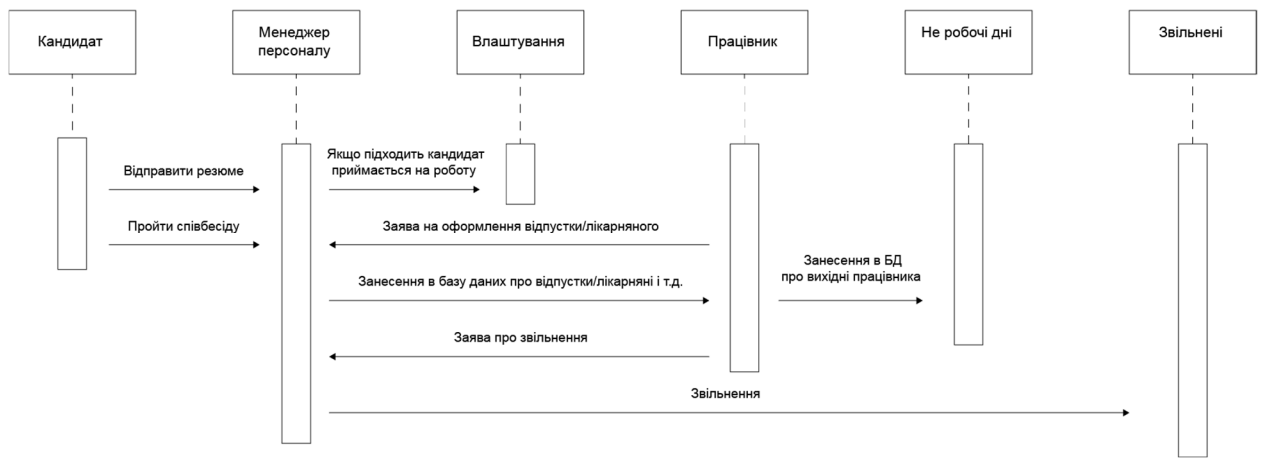


Рисунок 3.3 – Діаграма послідовності для відділу кадрів

Використовується для уточнення прецедентних діаграм – описує поведінкові сторони системи. Діаграма послідовності демонструє взаємодію об’єктів у часі. У цьому випадку інформація набуває форми повідомлень, а взаємодія об’єктів передбачає обмін цими повідомленнями всередині сценарію.

3.2 Проектування сховища даних відділу кадрів на основі SQL

MySQL Workbench – це уніфікований візуальний інструмент проектування баз даних або графічний інтерфейс користувача, що використовується для роботи з архітекторами баз даних, розробниками та адміністраторами баз даних. Він забезпечує розробку SQL, моделювання даних, міграцію даних та комплексні інструменти адміністрування для конфігурації сервера, адміністрування користувачів, резервного копіювання та багатьох інших. Ми можемо використовувати це адміністрування сервера для створення нових фізичних моделей даних та для розробки SQL (виконувати запити тощо). Він доступний для всіх основних операційних систем, таких як Mac OS, Windows та Linux [49].

Моделювання даних ця функціональність надає можливість створення

графічних моделей схеми бази даних, виконує зворотне та пряме проектування між Схемою та реальною базою даних та редагує всі аспекти бази даних за допомогою всеосяжного редактора таблиць. Редактор таблиць надає можливості для редагування таблиць, стовпців, індексів, подань, тригерів, розділення тощо [49].

Запускаємо MySQL Workbench та створимо нове підключення до БД. Назвемо підключення «departments» та використаємо локальний сервер 127.0.0.1 та стандартний порт 3306 (рисунок 3.4). Після підключення, відкривається вікно редактора SQL-запитів (рисунок 3.5).

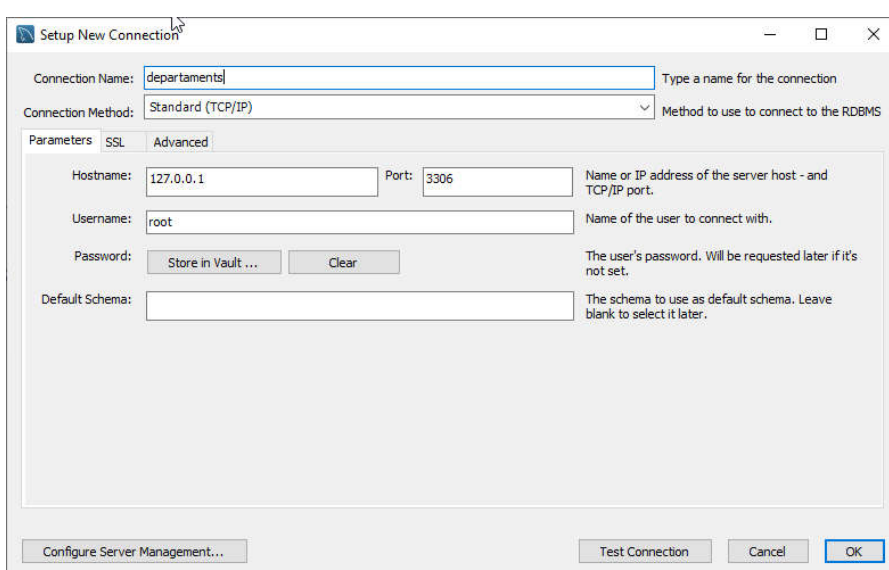


Рисунок 3.4 – Створення нового з'єднання

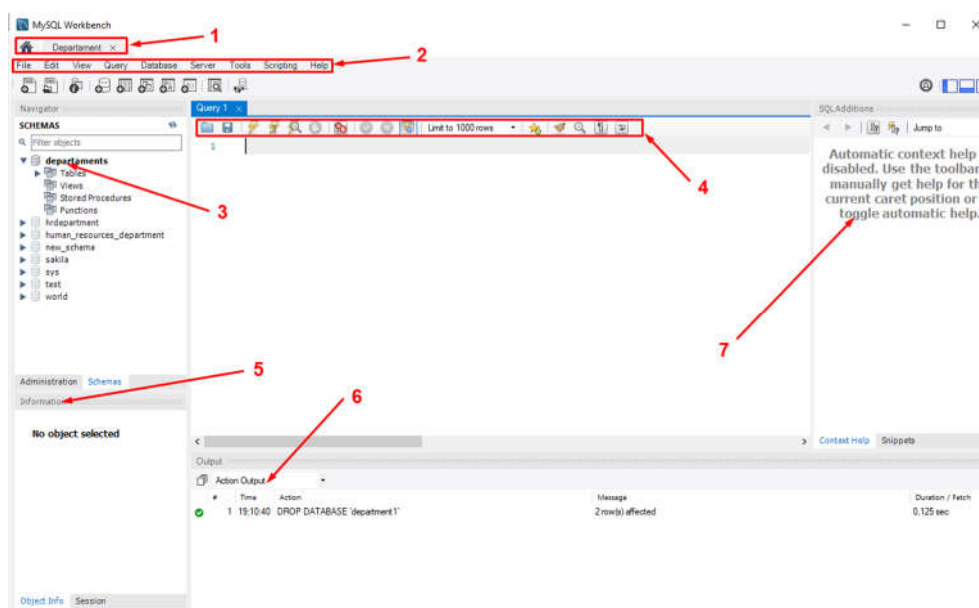


Рисунок 3.5 – Структура вікна редактора запитів

Змн.	Арк.	№ докум.	Підпис	Дата

Детальніше опишемо всі основні інструменти середовища розробки:

- 1) повернення до домашнього екрана і перемикання між вкладками;
- 2) меню список команд, у якому вибрати потрібну команду для виконання;
- 3) зліва знаходиться навігатор, що показує основні завдання та список об'єктів БД.;
- 4) вікно редагування запитів;
- 5) у лівому нижньому кутку знаходиться вікно інформації про вибраний об'єкт;
- 6) внизу вікно історії запитів та помилок;
- 7) справа знаходиться вікно допомоги.

На (рисунку 3.6) показано вікно моделювання EER діаграми для MySQLWorkbench.

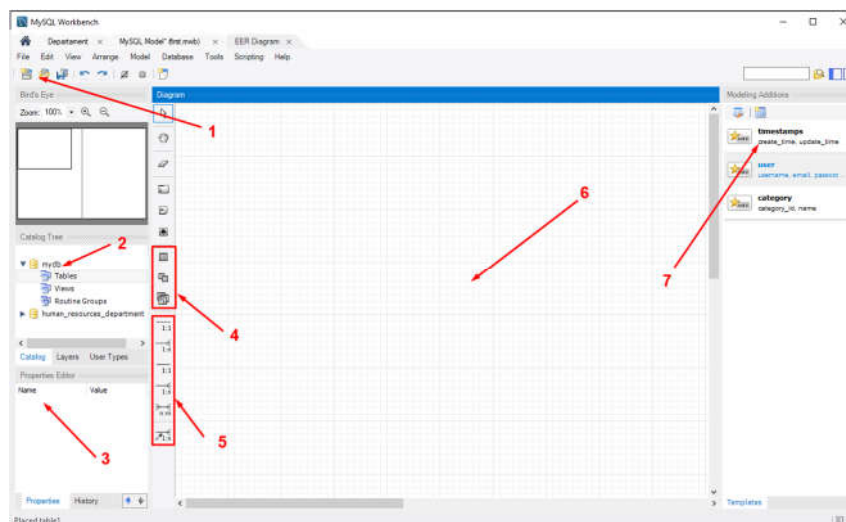


Рисунок 3.6 – Вікно створення схем БД

Нова вкладка з порожнім аркушем ER діаграми. Розглянемо можливості:

- 1) меню зберігання і створення нових об'єктів;
- 2) каталог з створеними схемами й таблицями;
- 3) вікно інформації про вибраний об'єкт;
- 4) інструменти створення нових таблиць;
- 5) інструменти для встановлення відношень;

- 6) робочий простір в якому розміщуються таблиці та зв'язки;
- 7) вікно шаблонів таблиць.

За допомогою програмного засобу MySQL Workbench створимо нову таблицю (рисунок 3.7) і назначимо типи даних в кінці заповнимо таблицю даними (рисунок 3.9) за допомогою графічного інтерфейсу програми [1].

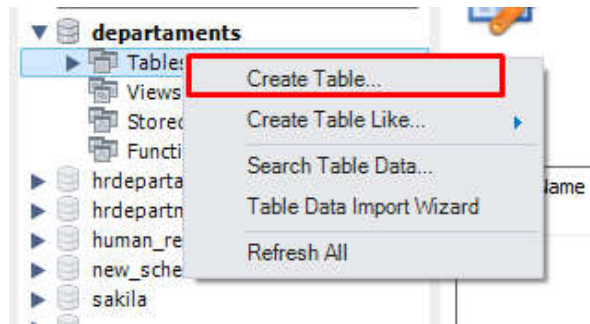


Рисунок 3.7 – Створення нової таблиці

Зверху вводимо назву таблиці «employee» зліва додаємо нові поля. Поле id задаємо як первинний ключ. Посередині назначимо типи даних (рисунок 3.8) [22].

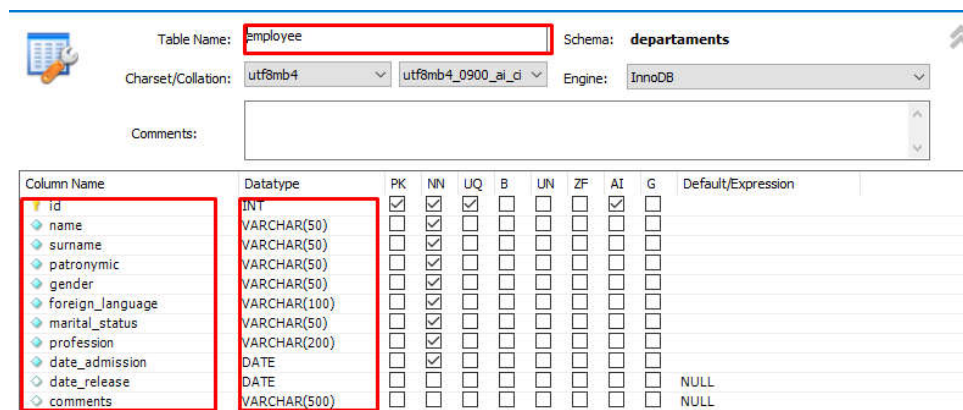


Рисунок 3.8 – Створення полів таблиці employee

Після натискання Apply створюється SQL-код для побудови нової таблиці (рисунок 3.9).

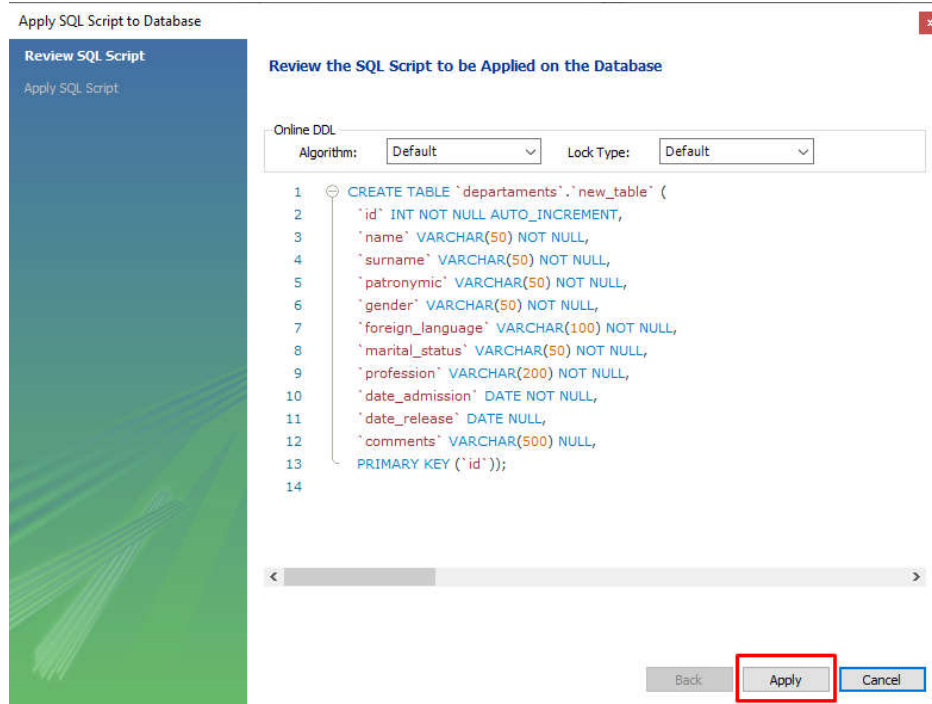


Рисунок 3.9 – SQL – код створення нової таблиці

Після застосування коду відкриємо таблицю для вводу даних (рисунок 3.10) та заповнимо її при цьому враховуємо, що поле id – має властивість автоінкременту. При виконанні згенерується код для додавання нових даних (рисунок 3.11).

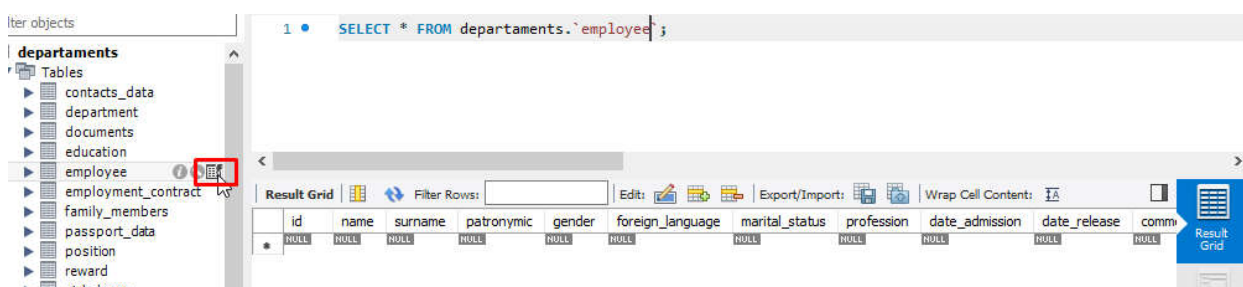


Рисунок 3.10 – Вікно редагування даних

Після вводу даних натискаємо кнопку Apply і з'являється автоматичне створення SQL коду (рисунок 3.11).

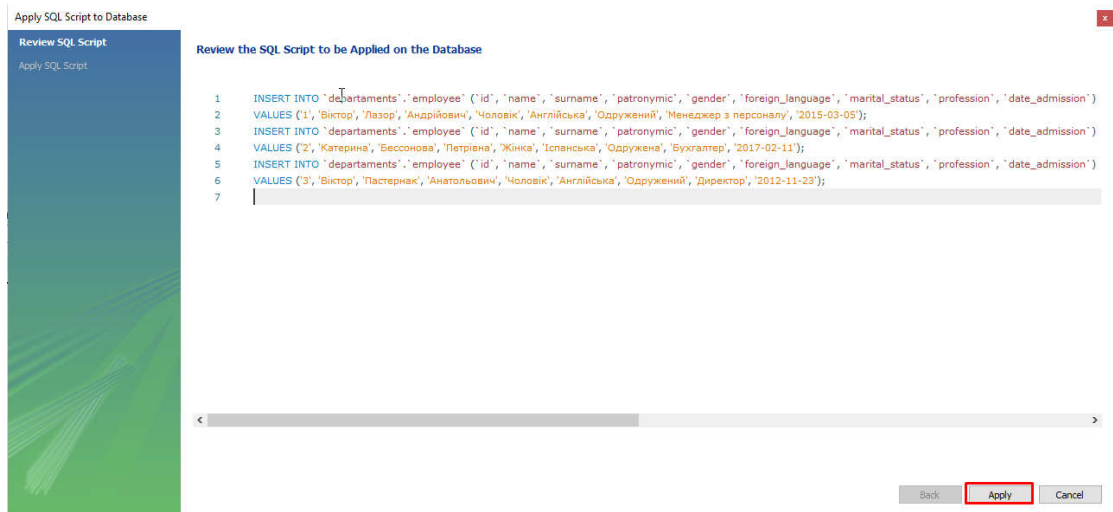


Рисунок 3.11 – SQL – код заповнення даними таблицю employee

Виконаємо декілька запитів на вибірку даних з таблиці employee (рисунок 3.12).

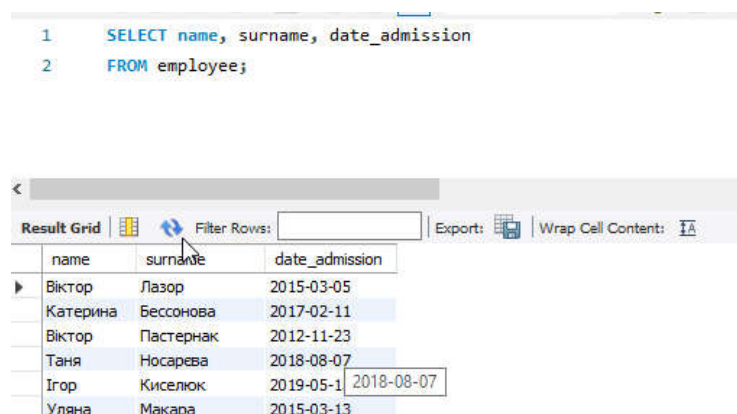


Рисунок 3.12 – Вибірка даних із таблиці employee

Використання програмного забезпечення рекомендується для складного переходу. У таблицях показані збережені процеси, зовнішні ключі. Підтримується інтегрована оболонка, яка дозволяє писати сценарії. Перш за все, програма є інструментом проектування візуальної графіки.

Існує редактор, який дозволяє коригувати запити на подальшу відправку через сервер. Прийняті відповіді подаються у вигляді таблиць. У візуальному вигляді користувач зберігає можливість вносити зміни.

3.3 Управління відділом кадрів на підприємстві на основі розробленого сховища даних

Кожна модель даних передбачає застосування певного математичного апарату та набору операцій над даними. Вони будуть показані нижче в прикладах робіт з об'єктами БД реляційного типу.

Перейдемо безпосередньо до створення бази даних. Для цього потрібно створити саму базу даних і таблиць відповідно до заданої схеми БД в СУБД MySQL можна використовувати запити, у середовищі MySQL Workbench (тут база даних називається department) [49].

Перед створенням запитів був встановлений і запущений MySQL Server, налаштоване підключення до цього сервера середовища MySQL Workbench. У базу були додані дані для перевірки коректності виконання запитів. Додавання проводилося за допомогою MySQL Workbench, в результаті було розроблено наступні запити:

Для створення бази даних використовується команда CREATE DATABASE. USE department – команда для вибору бази даних з якою будемо працювати.

Лістинг коду створення бази даних «department»

```
CREATE DATABASE department;  
USE department;
```

Лістинг коду створення таблиці «contacts_data»

```
CREATE TABLE `contacts_data` (  
  `id` int NOT NULL AUTO_INCREMENT,  
  `phone_number` varchar(13) NOT NULL,  
  `home_address` varchar(200) NOT NULL,  
  `email` varchar(50) NOT NULL,  
  PRIMARY KEY (`id`),  
  
  UNIQUE KEY `employee_id_UNIQUE` (`id`)
```

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		56

) ENGINE=InnoDB AUTO_INCREMENT=14 DEFAULT CHARSET=utf8mb4
 COLLATE=utf8mb4_0900_ai_ci

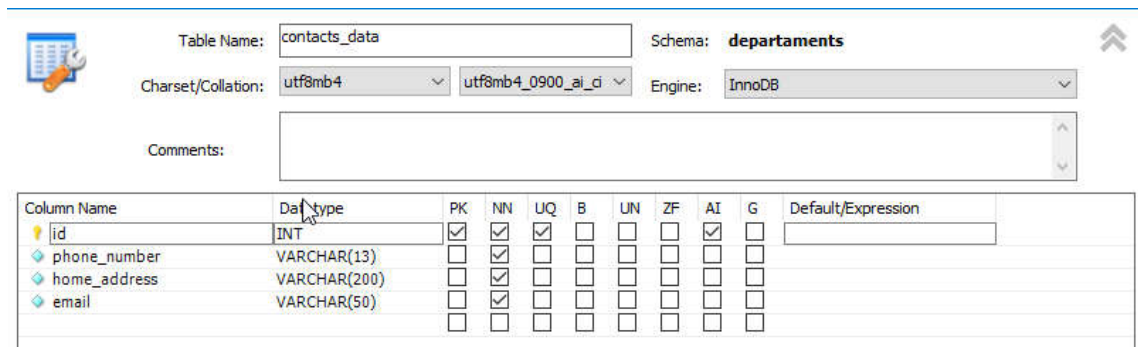


Рисунок 3.4 – Вигляд таблиці «contacts_data» в середовищі MySQL Workbench

Наведені запити успішно виконаний на сервері, в результаті – створені таблиці. Тут варто звернути увагу на іменування обмежень для зовнішніх ключів – вони повинні бути унікальними в базі даних, тому в імені обмеження кодується ім'я відносини, для якого його описано, наприклад:

PRIMARY KEY (`id`)

Видно, що для всіх ключових полів в базі даних проставлені індекси – внаслідок цього запису упорядковуються по цих полях і пошук виконується швидше (бінарний пошук замість лінійного).

Лістинг коду використання вводу даних в таблицю «contacts_data»:

```
LOCK TABLES `contacts_data` WRITE;
INSERT INTO `contacts_data` VALUES
(1, '+380672697273', 'м. Тернопіль вул. Заводська 7', 'victorlazor312@gmail.com'),
(2, '+380979000319', 'м. Тернопіль вул. Театральний 27', 'katerinabesonova31@gmail.com'),
(3, '+380982131532', 'м. Тернопіль вул. Бандери 21', 'viktorporaternak96@gmail.com'),
(4, '+380974125632', 'м. Тернопіль вул. Петлюри 12', 'tanyanasava52@gmail.com');
UNLOCK TABLES;
```


id	name	surname	patronymic	gender	foreign_language	marital_status	profession	date_admission	date_release	comments	department_id	passport_data_id	contacts_data_id
1	Віктор	Лазор	Андрійович	Чоловік	Англійська	Одружений	Менеджер з персоналу	2015-03-05	NULL	NULL	3	1	1
2	Катерина	Бессонова	Петрівна	Жінка	Іспанська	Одружена	Бухгалтер	2017-02-11	NULL	NULL	1	2	2
3	Віктор	Пастернак	Анатольович	Чоловік	Англійська	Одружений	Директор	2012-11-23	NULL	NULL	1	3	3
4	Таня	Носарева	Василівна	Жінка	Польська	Не одружена	Менеджер з продажу	2018-08-07	NULL	NULL	4	4	4
5	Ігор	Киселюк	Дмитрович	Чоловік	Російська	Не одружена	Адміністратор	2019-05-15	NULL	NULL	1	5	5
6	Уляна	Макара	Сергіївна	Жінка	Англійська	Одружена	Начальник відділу пр...	2015-03-13	NULL	NULL	1	6	6
7	Дмитро	Дорош	Євгенович	Чоловік	Італійська	Не одружена	Менеджер з розвитку	2014-07-09	NULL	NULL	3	7	7
8	Настя	Марущак	Андрівна	Жінка	Англійська	Не одружена	Спеціаліст з обліку	2019-05-28	NULL	NULL	4	8	8
9	Володимир	Гжовський	Петрович	Чоловік	Англійська	Одружений	ІТ-спеціаліст	2017-11-02	NULL	NULL	2	9	9
10	Тетяна	Яковенко	Богданівна	Жінка	Польська	Одружена	Начальник відділу ка...	2014-09-21	NULL	NULL	1	10	10
11	Христіна	Мигаль	Василівна	Жінка	Англійська	Одружена	Координатор	2013-10-12	NULL	NULL	4	11	11
12	Андрій	Новак	Стефанович	Чоловік	Англійська	Не одружений	Інженер	2018-11-19	NULL	NULL	2	12	12
13	Віталій	Дзюбинський	Ростиславович	Чоловік	Англійська	Одружений	Начальник ІТ відділу	2014-06-01	NULL	NULL	2	13	13

Рисунок 3.5 – Результат введення коду заповнення таблиці «contacts_data».

Код створення інших таблиць і вводу даних в них можна переглянути в (Додатку В).

Для вибірки даних в MySQL застосовується команда SELECT. Виконаємо операцію на вибірку даних з таблиці «employee» виведемо прізвище ім'я назву професії та в якому відділі вони працюють (рисунок 3.6):

Лістинг коду виводу даних з таблиці «employee» :

```
SELECT surname, name, profession,
       date_admission, department_id
FROM employee;
```

	surname	name	profession	date_admission	department_id
▶	Лазор	Віктор	Менеджер з персоналу	2015-03-05	3
	Бессонова	Катерина	Бухгалтер	2017-02-11	1
	Пастернак	Віктор	Директор	2012-11-23	1
	Носарева	Таня	Менеджер з продажу	2018-08-07	4
	Киселюк	Ігор	Адміністратор	2019-05-15	1
	Макара	Уляна	Начальник відділу пр...	2015-03-13	1
	Дорош	Дмитро	Менеджер з розвитку	2014-07-09	3
	Марущак	Настя	Спеціаліст з обліку	2019-05-28	4
	Гжовський	Володимир	ІТ-спеціаліст	2017-11-02	2
	Яковенко	Тетяна	Начальник відділу ка...	2014-09-21	1
	Мигаль	Христіна	Координатор	2013-10-12	4
	Новак	Андрій	Інженер	2018-11-19	2
	Дзюбинський	Віталій	Начальник ІТ відділу	2014-06-01	2

Рисунок 3.6 – Результат введення коду на вивід даних з таблиці «employee».

Найчастіше необхідно використовувати не всі дані з БД, а тільки ті, які відповідають певній умові. Для фільтрації даних в команді SELECT застосовується оператор WHERE, після якого вказується умова. Виконаємо операцію на вивід інформації про працівників які працюють в четвертому відділі (рисунок 3.7):

Лістинг коду виводу про працівників четвертого відділу

```
SELECT *  
FROM employee  
WHERE department_id = '4';
```

id	name	surname	patronymic	gender	foreign_language	marital_status	profession	date_admission	date_release	comments	department_id
4	Таня	Носарева	Басилівна	Жінка	Польська	Не одружена	Менеджер з продажу	2018-08-07	NULL	NULL	4
8	Настя	Марущак	Андріївна	Жінка	Англійська	Не одружена	Спеціаліст з обліку	2019-05-28	NULL	NULL	4
11	Христина	Мигаль	Басилівна	Жінка	Англійська	Одружена	Координатор	2013-10-12	NULL	NULL	4
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.7 – Результат введення коду на вивід даних про працівників четвертого відділу

Команда UPDATE застосовується для поновлення вже наявних рядків. Виконаємо зміну заробітної плати працівника з ідентифікатором один, одразу виконаємо перевірку змін (рисунок 3.8):

Лістинг коду зміни заробітної плати працівника

```
UPDATE `departments`.`employment_contract`  
SET `salary` = '15000.00' WHERE (`id` = '1');  
SELECT *  
FROM employment_contract  
WHERE id = '1';
```

id	date_of_conclusion	contract_series	contract_number	salary	release_order	reason_for_dismissal	employee_id
1	2015-03-05	HB	312413	15000.00	NULL	NULL	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.8 – Результат введення коду на зміну заробітної плати працівника

Оператор ORDER BY сортує значення по одному або декількох стовпцях. Наприклад, впорядкуємо вибірку з таблиці «employee» по даті початку роботи (рисунок 3.9):

Лістинг коду сортування по даті

```
SELECT * FROM employee  
ORDER BY date_admission;
```

	id	name	surname	patronymic	gender	foreign_language	marital_status	profession	date_admission
▶	3	Віктор	Пастернак	Анатольович	Чоловік	Англійська	Одружений	Директор	2012-11-23
	11	Христина	Мигаль	Василівна	Жінка	Англійська	Одружена	Координатор	2013-10-12
	13	Віталій	Дзюбинський	Ростиславович	Чоловік	Англійська	Одружений	Начальник IT відділу	2014-06-01
	7	Дмитро	Дорош	Євгенович	Чоловік	Італійська	Не одружена	Менеджер з розвитку	2014-07-09
	10	Тетяна	Яковенко	Богданівна	Жінка	Польська	Одружена	Начальник відділу ка...	2014-09-21
	1	Віктор	Лазор	Андрійович	Чоловік	Англійська	Одружений	Менеджер з персоналу	2015-03-05
	6	Уляна	Макара	Сергіївна	Жінка	Англійська	Одружена	Начальник відділу пр...	2015-03-13
	2	Катерина	Бессонова	Петрівна	Жінка	Іспанська	Одружена	Бухгалтер	2017-02-11
	9	Володимир	Гжовський	Петрович	Чоловік	Англійська	Одружений	IT-спеціаліст	2017-11-02
	4	Таня	Носарева	Василівна	Жінка	Польська	Не одружена	Менеджер з продажу	2018-08-07
	12	Андрій	Новак	Стефанович	Чоловік	Англійська	Не одружений	Інженер	2018-11-19
	5	Ігор	Киселюк	Дмитрович	Чоловік	Російська	Не одружена	Адміністратор	2019-05-15
	8	Настя	Марущак	Андрівна	Жінка	Англійська	Не одружена	Спеціаліст з обліку	2019-05-28
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3.9 – Результат введення коду на сортування по даті

При роботі з базою даних MySQL можна використовувати дані з декількох таблиць. Для об'єднання цієї інформації в мові SQL. При цьому мається на увазі, що при об'єднанні двох і більше таблиць, відносини між даними зберігаються. Об'єднаємо таблиці «employee», «employment_contract», «education», та виведемо інформацію про ім'я, прізвище, професію, дату підписання контракту, серію та номер контракту й заробітну плату (рисунок 3.10):

Лістинг коду об'єднання й вивід інформації з таблиць

```
SELECT name, surname, profession, date_of_conclusion,
contract_series, contract_number, qualification_level, salary
FROM employee, employment_contract, education
WHERE employee.id = employment_contract.id
AND employee.id = education.id;
```

	name	surname	profession	date_of_conclusion	contract_series	contract_number	qualification_level	salary
	Віктор	Лазор	Менеджер з персоналу	2015-03-05	НВ	312413	3	15000.00
▶	Катерина	Бессонова	Бухгалтер	2017-02-11	ЄФ	312412	6	15000.00
	Віктор	Пастернак	Директор	2012-11-23	ВЗ	153412	3	30000.00
	Таня	Носарева	Менеджер з продажу	2018-08-07	ХИ	645644	6	10000.00
	Ігор	Киселюк	Адміністратор	2019-05-15	Гь	714223	3	12000.00
	Уляна	Макара	Начальник відділу пр...	2015-03-13	ЗО	856755	6	20000.00
	Дмитро	Дорош	Менеджер з розвитку	2014-07-09	БА	532422	3	10000.00
	Настя	Марущак	Спеціаліст з обліку	2019-05-28	УН	764644	6	15000.00
	Володимир	Гжовський	IT-спеціаліст	2017-11-02	ЦТ	654343	3	15000.00
	Тетяна	Яковенко	Начальник відділу ка...	2014-09-21	ГЮ	462342	6	20000.00
	Христина	Мигаль	Координатор	2013-10-12	СР	875675	3	12000.00
	Андрій	Новак	Інженер	2018-11-19	ФП	873452	6	12000.00
	Віталій	Дзюбинський	Начальник IT відділу	2014-06-01	ЧЛ	655464	3	20000.00

Рисунок 3.10 – Результат введення коду виводу даних з різних таблиць

Виконані операції автоматично вносяться в базу даних і зберігаються на сервері. Такий спосіб зручний і простий в використанні. За допомогою команд SQL можна описати будь-які запити які чудово підходять для запису інформації про відділ кадрів підприємства.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		61

4.ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ ПРОГРАМНОГО ЗАСОБУ

У цьому розділі кваліфікаційної роботи здійснено економічне обґрунтування доцільності розробки програмного забезпечення для управління персоналом підприємства. Зокрема, розрахунок витрат на розробку цього програмного продукту, експлуатаційні витрати, ціни споживання проектного рішення, визначення показників економічної ефективності нового програмного продукту, обґрунтовуються відповідні висновки.

Розроблений програмний засіб для моделювання сховищ даних для управління персоналом підприємства призначений для зменшення навантаження на відділ кадрів для ведення обліку працівників.

4.1 Розрахунок витрат на розробку програмного забезпечення

Розробка нового програмного забезпечення вимагає свого контролю управління з боку керівника. Таким чином, складання та організація економічної частини є актуальною проблемою сучасного підприємства.

Для визначення загальної тривалості етапів розробки продукту сформовано таблицю 4.1.

Таблиця 4.1 – Стадії розробки програмного засобу

№ п/п	Назва операції (стадії)	Виконавець, посада	Середній час виконання операції, год.
1	Підготовка, складання ТЗ	Керівник проекту	4

2	Аналіз засобів моделювання	Студент	8
3	Створення моделей системи	Студент	16
4	Розробка системи	Студент	24
5	Тестування продукту	Студент	8
6	Встановлення та налаштування ПЗ	Студент	8
7	Представлення реалізованої системи	Керівник проекту	4
Всього		6	78

Розрахунок витрат на розробку і впровадження засобу управління кадрами підприємства (K) включають:

$$K = K_1 + K_2, \quad (4.1)$$

де K_1 – витрати на розробку програмного засобу, грн;

K_2 – витрати на експлуатацію та налагодження програмних задач на ПК, грн.

Витрати на розробку програмних засобів включають:

- витрати на оплату праці розробника ($B_{оп}$);
- витрати на відрахування у спеціальні державні фонди ($B_{ф}$);
- витрати на закупні вироби ($Пв$);
- витрати на придбання спец обладнання для проведення експериментальних робіт ($Об$);
- накладні витрати ($Н$);
- інші витрати ($Ів$).

Витрати на оплату праці включають заробітну плату (ЗП) усіх категорій працівників, безпосередньо зайнятих на всіх стадіях проектування. Розмір програмного забезпечення розраховується виходячи зі складності відповідної

роботи в людські дні та середнього рівня ЗП відповідних категорій працівників.

У розробці проектного рішення беруть участь такі спеціалісти-розробники, а саме: керівник проекту, студент, консультант техніко-економічного розділу (таблиця 4.2).

Таблиця 4.2 – Розмір заробітної плати виконавців

Посада виконавців	Місячний оклад, грн.
Керівник КР, старший викладач	7200
Консультант техніко-економічного розділу, старший викладач	7200
Студент	1400

Витрати на оплату праці розробників проекту визначаються за формулою:

$$B_{оп} = \sum_{i=1}^N \sum_{j=1}^M n_{ij} \cdot t_{ij} \cdot C_{ij}, \quad (4.2)$$

де n_{ij} – чисельність розробників i -ої спеціальності j -го тарифного розряду, осіб;

t_{ij} – затрачений час на розробку проекту співробітником i -ої спеціальності j -го тарифного розряду, год;

C_{ij} – годинна ставка працівника i -ої спеціальності j -го тарифного розряду, грн.

Середньогодинну ставку працівника розраховуємо за формулою:

$$C_{ij} = \frac{C_{ij}^0(1+h)}{PЧ_i}, \quad (4.3)$$

де C_{ij}^0 – основна місячна заробітна плата розробника i -ої спеціальності j -го тарифного розряду, грн.;

h – коефіцієнт, що визначає розмір додаткової заробітної плати (при умові наявності доплат);

$РЧ_i$ – місячний фонд робочого часу працівника i -ої спеціальності j -го тарифного розряду, год. (приймаємо 168 год.).

Результати розрахунку записуємо у таблицю 4.3, зважаючи на те, що коефіцієнт h для керівника і для консультанта з техніко-економічного розділу.

Таблиця 4.3 – Розрахунок витрат на оплату праці

Посада виконавців	Час розробки, год.	Погодинна заробітна плата, грн/год.	Витрати на розробку, грн
Керівник ДП, доцент	8	56	448
Консультант техніко-економічного розділу, доцент	2	42	84
Студент	70	8	560
Разом			1 092

Відрахування на соціальні заходи. Величну відрахувань у спеціальні державні фонди визначають у відсотковому співвідношенні від суми основної та додаткової заробітних плат. Згідно діючого нормативного законодавства сума відрахувань у спеціальні державні фонди складає 20,5% від суми заробітної плати:

$$V_{\phi} = \frac{20,5}{100} \cdot 980 = 193,68, \text{грн}$$

Розрахунок витрат на матеріали та комплектуючі. Загальна сума витрат на матеріальні ресурси (ВМ) визначається за формулою (4.4):

$$B_M = \sum_{i=1}^n K_i \cdot C_i \quad (4.4)$$

де K_i – витрата i -го типу матеріалу, натуральні одиниці вимірювання;

C_i – ціна за одиницю i -го типу матеріалу, грн.;

i – тип матеріального ресурсу;

n – кількість типів матеріальних ресурсів.

У таблиці 4.4 наведений перелік матеріалів та комплектуючих і розраховані витрати на них.

Таблиця 4.4 – Розрахунок витрат на матеріали та комплектуючі

№ п/п	Найменування купованих виробів	Одиниця виміру	Ціна, грн	Кількість купованих виробів	Сума, грн	Транспортні витрати (10% від суми)	Загальна сума, грн
1	Папір (формат А4)	уп.	90,0	2	180,00	18,0	198,0
2	Ручка кулькова	шт.	15,0	2	30,00	3,0	33,0
3	Олівець простий	шт.	6	1	6,00	0,6	6,6
4	Зошит, 96 арк.	шт.	36	1	36	3,6	39,6
5	Тонер для принтера	уп.	100	1	100	10,0	110,0
Разом							387,2

Витрати на використання комп'ютерної техніки. Витрати на використання комп'ютерної техніки (ВКТ) включають витрати на амортизацію комп'ютерної техніки, витрати на користування програмним забезпеченням, витрати на електроенергію, що споживається комп'ютером. За даними обчислювального центру ЗУНУ для комп'ютера типу IBM PC/ATX вартість години роботи становить 6 грн. Середній щоденний час роботи на комп'ютері

– 2 години. Розрахунок витрат на використання комп'ютерної техніки приведений в таблиці 4.5.

Накладні витрати включають три групи витрат: витрати на управління, загальновиробничі витрати, невиробничі витрати. Вони розраховуються у відсотках до витрат на оплату праці.

Таблиця 4.5 – Розрахунок витрат на використання комп'ютерної техніки

№ п/п	Назва етапів робіт, при виконанні яких використовується комп'ютер	Час використання комп'ютера, год.	Витрати на використання комп'ютера, грн.
1	Проведення досліджень та оформлення їх результатів	70	420
2	Оформлення техніко-економічного розділу	10	60
4	Оформлення БР	24	144
Разом		84	624

Середньостатистичний відсоток накладних витрат приймемо від 0,7 до 100 % від заробітної плати:

$$H = 1,5 \cdot 980 = 1\,470, \text{ (грн.)}$$

Інші витрати є витратами, які не враховані в попередніх статтях. Вони становлять 10% від заробітної плати:

$$I = 980 \cdot 0,1 = 98, \text{ (грн.)}$$

Витрати на розробку програмного забезпечення складають:

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		67

$$K_1 = B_{ОП} + B_{\Phi} + B_M + H + I, \quad (4.4)$$

$$K_1 = 980 + 193,68 + 717,2 + 1\,470 + 98 = 3\,458,88 \text{ (грн.)}$$

Витрати на владження і дослідну експлуатацію програмного продукту визначаємо за формулою:

$$K_2 = S_{м.г.} \cdot t_{від}, \quad (4.5)$$

де $S_{м.г.}$ – вартість однієї машино-години роботи ПК, грн./год;

$t_{від}$ – комп'ютерний час, витрачений на владження і дослідну експлуатацію створеного програмного продукту, год.

Загальна кількість днів роботи на комп'ютері дорівнює 28 днів. Середній щоденний час роботи на комп'ютері – 3 години. Вартість години роботи комп'ютера дорівнює 6 грн., тому:

$$K_2 = 6 \cdot 28 \cdot 3 = 504 \text{ грн.}$$

4.2 Розрахунок експлуатаційних витрат і ціни

Для оцінки економічної ефективності розробленого програмного продукту слід порівняти його з аналогом, тобто існуючим програмним забезпеченням ідентичного функціонального призначення. Для цього визначимо експлуатаційні витрати на розробку проекту.

Експлуатаційні одноразові витрати по програмному забезпеченню і аналогу включають вартість підготовки даних і вартість роботи комп'ютера (за час дії програми):

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		68

$$E_{\Pi} = E_{1\Pi} + E_{2\Pi}, \quad (4.6)$$

де E_{Π} – одноразові експлуатаційні витрати на ПЗ (аналог), грн.;
 $E_{1\Pi}$ – вартість підготовки даних для експлуатації ПЗ (аналогу), грн.;
 $E_{2\Pi}$ – вартість роботи комп'ютера для розробки програмного продукту (аналогу), грн.

Річні експлуатаційні витрати $B_{E\Pi}$ визначаються за формулою:

$$B_{E\Pi} = E_{\Pi} \cdot N_{\Pi}, \quad (4.7)$$

де N_{Π} – періодичність експлуатації ПЗ (аналогу), раз/рік.

Вартість підготовки даних для роботи на комп'ютері визначається за формулою:

$$E_{1\Pi} = \sum_{i=1}^n n_i t_i c_i, \quad (4.8)$$

де i – категорії працівників, які приймають участь у підготовці даних ($i=1,2,\dots,n$);

n_i – кількість працівників i -тої категорії, осіб;

t_i – трудомісткість роботи співробітників i -тої категорії по підготовці даних, год.;

c_i – середньо годинна ставка працівника i -тої категорії з врахуванням додаткової заробітної плати, що знаходиться із співвідношення (4.3):

$$C_i = \frac{C_i^o(1+b)}{m}, \quad (4.9)$$

де C_i^o – основна місячна заробітна плата працівника i -ої категорії, грн;

b – коефіцієнт, який враховує додаткову заробітну плату (приймемо 0);

m – кількість робочих годин у місяці, год.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
						69
Змн.	Арк.	№ докум.	Підпис	Дата		

Для роботи з даними як для проектного рішення так і аналогу потрібен один працівник, основна місячна заробітна плата якого складає: $c = 7\,446$ грн.

Тоді:

$$C_i = \frac{7\,446(1+0)}{22 \cdot 8} = 42,30 \text{ грн/год}$$

Трудомісткість підготовки даних для даного проектного рішення складає 2 год., відповідно для аналога – 3 год. Результати представлені у таблиці 4.6.

Таблиця 4.6 – Розрахунок витрат на підготовку даних та реалізацію проектного рішення на комп'ютері

Час роботи співробітників, год.	Середньо годинна заробітна плата, грн./год.	Витрати , грн.
Проектне рішення		
2	42,30	84,6
Аналог		
3	42,30	126,9

Витрати на експлуатацію комп'ютера визначаються за формулою:

$$E_{2П} = t * S_{МГ} , \quad (4.10)$$

де t – витрати машинного часу для реалізації проектного рішення (аналогу), год.;

$S_{МГ}$ – вартість однієї години роботи комп'ютера, грн./год.

$$E_{2П} = 1 \cdot 6 = 6 \text{ грн.}, E_{2Па} = 1,5 \cdot 6 = 9 \text{ грн.}$$

$$E_{П} = 84,6 + 6 = 90,6 \text{ грн.}, E_{А} = 23,85 + 9 = 135,9 \text{ грн.}$$

					КР.КІ.07091/17.00.00.000 ПЗ	Арк. 70
Змн.	Арк.	№ докум.	Підпис	Дата		

$$E_{\text{ЕП}} = 84,6 \cdot 252 = 21\,319,2 \text{ грн.}, E_{\text{А}} = 135,9 \cdot 252 = 34\,246,8 \text{ грн.}$$

Накладні витрати пов'язані з обслуговуванням виробництва, утриманням апарату управління підприємства (фірми) та створення необхідних умов праці.

В залежності від організаційно-правової форми діяльності господарюючого суб'єкта, накладні витрати можуть становити 60–100 % від суми основної та додаткової заробітної плати працівників.

$$H_{\text{В}} = 0.7 \cdot V_{\text{ОП}} \quad (4.11)$$

де $H_{\text{В}}$ – накладні витрати.

$$H_{\text{В}} = 0.7 \cdot 980 = 686 \text{ грн.}$$

Складання кошторису витрат та визначення собівартості. Результати проведених розрахунків зведемо у таблицю 4.6

Таблиця 4.6 – Кошторис витрат на розробку програмного забезпечення

№ п/п	Найменування витрат	Сума витрат, грн.
1	Витрати на оплату праці	980
2	Відрахування у спеціальні державні фонди	193,68
3	Витрати на куповані вироби	717,2
4	Накладні витрати на розробку	1470
5	Інші витрати	98
6	Витрати на владження і дослідну експлуатацію програмного продукту	420
7	Накладні витрати експлуатацію	686
8	Річні експлуатаційні витрати	34246,8
Разом		38 811

4.3 Визначення показників економічної ефективності

Для того, щоб побудувати таблицю показників економічної ефективності розробки програмного продукту, проведемо розрахунки необхідних показників.

Для початку давайте визначимо прогнозовану ціну.

Величина можливої (договірної) ціни повинна визначатися з урахуванням ефективності, якості і термінів її виконання на рівні, що відповідає економічним інтересам замовника (споживача) і виконавця (підрядника). Договірна ціна (C_D) для прикладних КС розраховується за формулою:

$$C_D = B_{KC} \cdot \left(1 + \frac{p}{100}\right), \quad (4.12)$$

де B_{KC} – кошторисна вартість, грн.;

p – середній рівень рентабельності, (приймаємо 20-30%)

$$C_D = 38\,811 \cdot (1 + 0.26) = 48\,901,86 \text{ грн.}$$

Економічна ефективність (E_Φ) полягає у відношенні результату виробництва до затрачених ресурсів:

$$E_\Phi = \frac{\Pi}{B_{KC}}, \quad (4.13)$$

де Π – прибуток, грн.;

B_{KC} – кошторисна вартість, грн..

Поряд із економічною ефективністю розраховують термін окупності капітальних вкладень (T_p):

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		72

$$T_p = \frac{1}{E_\Phi}$$

$$\Pi = 48\,901,86 - 38\,811 = 10\,090,86$$

$$E_\Phi = \frac{10\,090,86}{48\,901,86} = 0,2063$$

$$T_p = \frac{1}{0,2063} = 4,84$$

Розраховані економічні показники щодо розробки проекту заносимо у таблицю 4.7

Таблиця 4.7 Розрахунок економічні показники

Показник	Значення
Ціна, грн.	48 901,86
Собівартість, грн.	38 811
Плановий прибуток, грн.	10 090,86
Економічна ефективність	0,2063
Термін окупності, рік	4,84

Враховуючи основні економічні показники з таблиці 4.7, можна зробити висновок, що при економічній ефективності 0,2063 та терміні окупності – 4,84 роки проводити роботи по впровадженню даного програмного модуля є доцільним та економічно вигідним

ВИСНОВКИ

1. Проведене дослідження особливостей функціонування відділу кадрів, що дало можливість проаналізувати існуючі методики оцінки ефективності функціонування підприємств та виявлення недоліків у роботі розробки засобу з впровадженням сучасних підходів, які ґрунтуються на впровадженні інформаційних технологій.

2. Проведено аналіз засобів управління базами даних, що дозволило проаналізувати представлені технології автоматизації систем управління інформацією та провести оцінку існуючих програмних засобів та було обрано реляційний тип баз даних.

3. Проведено розгляд можливостей реляційних сховищ даних для управління відділом кадрів що дозволяє зберігати велику кількість даних в таблицях які поєднуються між собою зв'язками. Тобто зміна даних в одній таблиці міняє дані в інших.

4. Спроектовано концептуальну, логічну та фізичну як і будь-який процес проектування баз даних та інформаційної системи починається з високого рівня абстракції й стає все більш конкретним. Тому було створено концептуальну модель, що переходить до логічної моделі та завершиться фізичною моделлю.

5. В результаті робіт змодельована архітектура сховищ даних для управління кадрами підприємства, виконано проектування бази даних та основних таблиць, що забезпечать ефективне збереження даних та дозволять максимально ефективно опрацьовувати внесені дані.

6. Розроблено засоби управління відділом кадрів на підприємстві на основі реляційних сховищ даних, що надає доступ до баз даних і управління вхідною та вихідною інформацією. Це дає змогу швидко виконувати операції з даними та зберігати їх на сервері за допомогою SQL команд.

7. Останнім етапом виконання бакалаврської кваліфікаційної роботи було проведення техніко-економічний аналіз розробки програмного засобу. За

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		74

допомогою цього аналізу продемонстровано економічну ефективність розробленого програмного засобу та обґрунтовано вартість розробки.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		75

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. С.А. Мартишин, В.Л. Симонов, М.Б. Храпченко. Проектування і реалізація баз даних в СУБД MySQL з впровадженням MySQL Workbench. Москва, 2019. 160 с.
2. Петрович Й.М. Управління діяльністю організаційно-виробничих систем. Київ, 2013. 510 с.
3. Берко А. Ю. Верес О. М. Пасічник В. В. Системи управління базами даних та знань. Львів, 2006. 584 с.
4. Database [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://en.wikipedia.org/wiki/Database>.
5. Вендров А.М. Практикум з проектування програмного забезпечення економічних інформаційних систем: Учб. посібник / А.М. Вендров. – М.: Фінанси і статистика, 2004. – 192 с.
6. Васильков В. Г. Організація виробництва: Навч. посіб. / В. Г. Васильков – К. : КНЕУ, 2003. – 524 с.
7. Microsoft SQL Server [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server.
8. Єр'оміна Н.В. Проектування баз даних: Навч. посібник. – К.: КНЕУ, 1998. – 208 с.
9. Гайна Г. А. Основи проектування баз даних: навч. посібник / Г. А. Гайна – К. : КНУБА, 2005. – 204 с.
10. UML [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/UML>.
11. Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті: Навч. посібник. – Тернопіль: Карт-бланш, 2001. – 354с.
12. Національного університету «Львівська політехніка», 2010. – 260 с.
13. Адамчук С.В. Актуалізація кадрового потенціалу як чинник професіоналізації державної служби. Економіка & держава. 2014. №7. С. 137-140.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		76

14. Карпова Т.С. Базы данных: модели, разработка, реализация / Т.С. Карпова. – СПб.: Питер, 2001. – 304 с.
15. Конгаловский М.Р. Энциклопедия технологий баз данных. – М.: Финансы и статистика, 2002. – 800 с.
16. Консолидація інформації та інформаційна безпека [Електронний ресурс] / Н. Е. Кунанець, В. В. Пасічник // Системи обробки інформації. – 2010. – Вип. 3. – С. 131-133. – Режим доступу: http://nbuv.gov.ua/j-pdf/soi_2010_3_56.pdf
17. Корнеєв В.В. Базы данных. Интеллектуальная обработка информации / В.В. Корнеєв, А.Ф. Гарєєв, С.В. Васютин, В.В. Райх. – М.: Издавец Молгачева С.В., Издавництво Нолидж, 2001, - 496 с.
18. Корнієнко С. К. Системи баз даних: організація та проектування: Навчальний посібник / С. К. Корнієнко. – Запоріжжя : ЗНТУ, 2006. – 252 с.
19. Марков О.С. Базы данных. Введение у теорію і методологію: підручник / О.С. Марков, К.Ю. Лисовкий. – М.: Финансы и статистика, 2004. – 512 с.
20. Основы інформаційних систем: Навч. посібник. — 2-ге вид., перероб. і допов. / В. Ф. Ситник, Т. А. Писаревська, Н. В. Єрьоміна, О. С.Краєва; За ред. В.Ф. Ситника. – К. : КНЕУ, 2001. – 420 с.
21. Пасічник В. В. Сховища даних: Навчальний посібник / В. В. Пасічник, Н.Б. Шаховська. – Львів : Магнолія, 2008. – 496 с.
22. Пасічник В. В. Глобальні інформаційні системи та технології (моделі ефективного аналізу, опрацювання та захисту даних) / В.В. Пасічник,
23. Martin Kleppmann. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems / Martin Kleppmann. – 41 E University Ave, Champaign, IL 61820,: O'Reilly Media; 1 edition. – 616 с.
24. Jamie Chan. SQL: Learn SQL (using MySQL) in One Day and Learn It Well. SQL for Beginners with Hands-on Project. / Jamie Chan. – 2711 Centerville Road Suite 400 Wilmington, DE 198: Amazon Digital Services LLC. – 166 с.
25. PostgreSQL Wiki [Електронний ресурс] // [postgresql.org](https://wiki.postgresql.org) – Режим доступу до ресурсу: https://wiki.postgresql.org/wiki/Main_Page

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		77

26. Introduction to the Oracle Database [Електронний ресурс] // Oracle. – 2017. – Режим доступу до ресурсу: https://docs.oracle.com/cd/B19306_01/server.102/b14220/intro.htm
27. Microsoft SQL Server [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: https://ru.wikipedia.org/wiki/Microsoft_SQL_Server.
28. MySQL [Електронний ресурс] // Wikipedia. – 2017. – Режим доступу до ресурсу: <https://ru.wikipedia.org/wiki/MySQL>.
29. Карпова Т.С. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2001. – 304 с.
30. Л. В. Рудикова. Базы данных. Разработка приложений. – БХВПетербург, 2006. – 469 с.
31. Завадський І.О. Основи баз даних. – К.: Видавець, 2011. – 192 с.
32. Д. Кренке. Теория и практика построения баз данных. – СПб.: Питер, 2005. – 864 с.
33. Гураль І.В., Дубчак Л.О. Методичні вказівки до оформлення курсових проектів, звітів про проходження практики, випускних кваліфікаційних робіт для студентів спеціальності «Комп'ютерна інженерія»/Під ред. О.М. Березького. Тернопіль: ТНЕУ, 2019. 33 с.
34. Адамчук С.В. Актуалізація кадрового потенціалу як чинник професіоналізації державної служби. Економіка & держава. 2014. №7. С. 137-140.
35. Вовчак І.С. Інформаційні системи та комп'ютерні технології в менеджменті: Навч. посібник. – Тернопіль: Карт-бланш, 2001. – 354с
36. Гайна Г.А. Основи проектування баз даних: Навчальний посібник / Г.А. Гайна. – К.: КНУБА, 2005. –204 с
37. Дрешпак В.М. Сучасні напрями зарубіжних досліджень у сфері електронного урядування (на прикладі публікацій міжнародного журналу Electronic Government). Аспекти публічного управління. 2017. № 5-6. С. 17 - 26.
38. . Завадський І.О. Основи баз даних: [Навч. посіб.] / І.О. Завадський. — К.: Видавець І.О. Завадський, 2014. —192 с.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		78

39. Зарицька О.Л. Бази даних та інформаційні системи: [Метод. пос.] / О.Л. Зарицька. –Житомир: Вид-во ЖДУ ім. І. Франка, 2009. –132 с.
40. Кучук Г.А. Формалізація предметної області багатовимірних баз даних / Г.А. Кучук // Системи обробки інформації. – 2001. – № 1. – С. 110-114.
41. Основи інформаційних систем: Навч. посібник / Ситник В.Ф., Писаревська Т.А., Єршоміна Н.В., Краєва О.С.; Заред. Ситника В.Ф. – К.: КНЕУ, 2001. – 420 с.
42. Методичні вказівки до виконання курсової роботи з дисципліни «Технологія проектування та адміністрування баз даних і сховищ даних» для 110 студентів заочної форми навчання спеціальності 6.050102 „Економічна кібернетика” / Укладач: О.В. Шпортько.– Рівне: РДГУ, 2011.
43. Шаховська Н. Б. Сховища та простори даних: монографія / Н. Б. Шаховська, В. В. Пасічник; Міністерство освіти і науки України, Національний університет “Львівська політехніка”. – Львів: Видавництво Національного університету “Львівська політехніка”, 2009. – 240 с
44. Кибанов А.Я. Основи управління персоналом - М.: ИНФРА-М, 2006. - 304 с.
45. Карташова А.В. Управління людськими ресурсами - М.: ИНФРА-М, 2005. - 236 с.
46. Зінченко С. Управління персоналом як особливий вид управлінської діяльності // Вісник Національної академії державного управління при Президентові України. -2005. -№ 3. - С. 156-161 с.
47. Комісарова Т.А. Управління людськими ресурсами: навч. посібник - М.: Справа, 2002. - 312 с.
48. Кушплер Л. Практика управління людськими ресурсами на українських підприємствах. // Синергія. -2004. -№ 1. - С. 4- 10с
49. MySQL Workbench Manual. [Електронний ресурс] – 08.12.2019 – Режим доступу: <https://dev.mysql.com/doc/workbench/en/wb-intro.html>.
50. Scott W. Amble. The Elements of UML(TM) 2.0 Style / Scott W. Amble.. – 202 с.

					КР.КІ.07091/17.00.00.000 ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		79