

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ
ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНИХ НАУК**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Інформаційна система для управління абонементом
спортивно-оздоровчого комплексу на мові програмування
JavaScript з використанням Firebird Database»

на здобуття освітнього ступеня бакалавра
зі спеціальності 122 Комп'ютерні науки
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерні науки
(назва)

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис)

Богдан ПРОКАЗА
(Ім'я, ПРИЗВИЩЕ здобувача)

Виконав: здобувач вищої освіти гр. КНД-42
Богдан ПРОКАЗА

Керівник:
науковий ступінь,
вчене звання

Сергій ПРОКОПОВ
к.т.н., доцент

Рецензент:
науковий ступінь,
вчене звання

(Ім'я, ПРИЗВИЩЕ)

Київ 2025

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ**
Навчально-науковий інститут інформаційних технологій

Кафедра Комп'ютерних наук

Ступінь вищої освіти Бакалавр

Спеціальність 122 Комп'ютерні науки

Освітньо-професійна програма Комп'ютерні науки

ЗАТВЕРДЖУЮ

Завідувач кафедри Комп'ютерних наук

_____ Віктор ВИШНІВСЬКИЙ
« _____ » _____ 2025 р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Проказі Богдану Ігоровичу
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: «Інформаційна система для управління абонементом спортивно-оздоровчого комплексу на мові програмування JavaScript з використанням Firebird Database»

керівник кваліфікаційної роботи Сергій ПРОКОПОВ к.т.н., доцент,
(Ім'я, ПРИЗВИЩЕ науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «24» 02.2025 р. № 56

2. Строк подання кваліфікаційної роботи «30» травня 2025 р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, мови програмування HTML, CSS, JavaScript, Firebird Database, параметри для створення інформаційної системи.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Аналіз інформаційних систем, їх класифікація, архітектура та створення.
Дослідження інформаційних процесів при контролі абонементів спортивно-оздоровчого комплексу.

Розробка інформаційної системи для контролю абонементів спортивно-оздоровчого комплексу.

5. Перелік графічного матеріалу: *презентація*

1. Схема меню інформаційної системи.
2. Створення таблиць, ключів, процедур бази даних.
3. Діаграма бази даних.
4. Екранні форми інформаційної системи.
5. Фрагменти програмного коду.

6. Дата видачі завдання «25» лютого 2025 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз наявної науково-технічної літератури	25.02-09.03.25	виконано
2	Аналіз та дослідження інформаційних систем	09.03-16.03.25	виконано
3	Дослідження інформаційних процесів, що виникають при роботі спортивно-оздоровчого комплексу	17.03-23.03.25	виконано
4	Вибір інструментальних засобів для розробки інформаційної системи	24.03-29.03.25	виконано
5	Створення бази даних для інформаційної системи	01.04-07.04.25	виконано
6	Розробка інформаційної системи для спортивно-оздоровчого комплексу	08.04-24.04.25	виконано
7	Оформлення роботи: вступ, висновки, реферат	25.04-02.05.25	виконано
8	Розробка демонстраційних матеріалів	03.05-08.05.25	виконано

Здобувач вищої освіти

(підпис)

Богдан ПРОКАЗА

(Ім'я, ПРІЗВИЩЕ)

Керівник

кваліфікаційної роботи

(підпис)

Сергій ПРОКОПОВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 61 стор., 5 табл., 47 рис., 43 джерела.

Мета роботи – розробити інформаційну систему для обліку та контролю абонементів в спортивно-оздоровчому комплексі, призначену для надання оперативної інформації для обліку і подальшого її аналізу та зниження обсягів паперового документообігу.

Об'єкт дослідження – спортивно-оздоровчий комплекс.

Предмет дослідження – інформаційна система для контролю абонементів в спортивно-оздоровчому комплексі.

Короткий зміст роботи:

Зроблено аналіз інформаційних систем, вивчені їх класифікація, архітектура та створення.

Досліджено інформаційні процеси при контролі абонементів спортивно-оздоровчого комплексу.

За допомогою технологій HTML, CSS, JavaScript для створення клієнтської частини та технологій Node.js, AJAX, Firebird Database для створення серверної частини розроблена інформаційна система для контролю абонементів спортивно-оздоровчого комплексу, призначена для надання оперативної інформації для обліку та подальшого аналізу.

Розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі включає:

- створення бази даних для інформаційної системи;
- створення форм для надання нормативно-довідникової інформації;
- створення вхідних та вихідних документів.

Ключові слова: інформаційна система, спортивно-оздоровчий комплекс, облік та контроль абонентів, JavaScript, Node.js, AJAX, Firebird Database

ABSTRACT

Text part of the bachelor's qualification work: 61 pages, 5 tables, 47 pictures, 43 sources.

The purpose of the work is to develop an information system for accounting and controlling subscriptions in a sports and recreation complex, designed to provide operational information for accounting and further analysis and reduce the volume of paper document flow.

The object of the study – the sports and recreation complex.

The subject of the study – an information system for controlling subscriptions in a sports and recreation complex.

Summary of the work:

An analysis of information systems was made, their classification, architecture and creation were studied.

Information processes in controlling subscriptions in a sports and recreation complex were studied.

Using HTML, CSS, JavaScript technologies to create the client part and Node.js, AJAX, Firebird Database technologies to create the server part, an information system for controlling subscriptions in a sports and recreation complex was developed, designed to provide operational information for accounting and further analysis.

Development of an information system for accounting and control of subscriptions in a sports and recreation complex includes:

- creation of a database for the information system;
- creation of forms for providing regulatory and reference information;
- creation of input and output documents.

Keywords: information system, sports and recreation complex, accounting and control of subscribers, JavaScript, Node.js, AJAX, Firebird Database

ЗМІСТ

ВСТУП.....	10
1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	13
1.1 Інформаційні системи.....	13
1.2 Класифікація та створення інформаційних систем	15
1.3 Дворівнева, трирівнева та розподілена однорангова архітектура інформаційної системи	18
2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЦЕНТРОМ.....	23
2.1 Вибір бази даних для інформаційної системи	23
2.2 Вибір мови програмування клієнтської частини	25
2.2.1 HTML (Hypertext Markup Language)	26
2.2.2 CSS (Cascading Style Sheets)	28
2.2.3 JavaScript.....	32
2.2.4 Взаємодія між HTML, CSS та JavaScript	37
2.3 Організація інтеграції клієнтської та серверної частин	38
3 ОПИС ІНФОРМАЦІЙНОЇ СИСТЕМИ.....	40
3.1 Структура меню інформаційної системи.....	40
3.2 Опис системи контролю абонементів спортивно-оздоровчого комплексу ...	40
3.3 Логічна структура бази даних.....	41
3.3.1 Нормативно-довідникова інформація.....	41
3.3.2 Вхідна інформація.....	42
3.4 Створення бази даних	43
3.4.1 Створення таблиць бази даних.....	43
3.4.2 Створення ключів	45
3.4.3 Створення процедур	47
3.4.4 Створення діаграми бази даних.....	55
3.5 Екранні форми інформаційної системи	57
3.5.1 Нормативно-довідникова інформація.....	57
3.5.2 Вхідна інформація.....	61

3.5.3 Вихідна інформація	66
ВИСНОВКИ.....	68
ФРАГМЕНТИ ПРОГРАМНОГО КОДУ.....	73
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ	94

ВСТУП

Однією з ключових умов ефективного розвитку сучасного підприємства є здатність оперативно реагувати на зміни ринкового середовища. Рациональне використання інформації як стратегічного ресурсу дає змогу контролювати всі аспекти діяльності компанії, своєчасно виявляти слабкі місця, зосереджувати зусилля на критичних напрямках, а також формувати найоптимальніші шляхи управління й зростання бізнесу. Для реалізації цього необхідно мати у розпорядженні спеціалізовані інструменти, що забезпечують збір, збереження та аналітичну обробку даних.

У сучасному світі інформаційні потоки досягають колосальних обсягів і постійно зростають. Це створює виклик для будь-якої організації – незалежно від її розміру – щодо ефективного управління даними.

Попри те, що деякі компанії досі користуються паперовими архівами, більшість уже перейшла до цифрових рішень – баз даних, які дозволяють упорядковувати, структурувати та надійно зберігати великі масиви інформації. Сьогодні бази даних стали невід’ємною частиною функціонування фінансових установ, промислових підприємств, торговельних організацій та багатьох інших сфер. Без них сучасна компанія не змогла б впоратися з лавиною інформації, що щоденно на неї обрушується.

Існує чимало переконливих причин для переходу на комп’ютеризовану обробку інформації. Зараз зберігання даних у цифровому форматі обходиться значно дешевше, ніж традиційне паперове. Бази даних надають можливість не лише зберігати великі обсяги інформації, а й впорядковувати її та швидко отримувати потрібні відомості у зручній для користувача формі.

Сучасні клієнт-серверні технології дозволяють економити як матеріальні ресурси, так і час, необхідний для доступу до інформації. Вони спрощують процес обробки та ведення даних, оскільки ґрунтуються на централізованому зберіганні й комплексному аналізі інформаційних масивів [1].

Крім того, персональні комп'ютери дають змогу працювати з даними у найрізноманітніших форматах – текстових документах, графічних зображеннях, рукописах, аудіофайлах тощо.

Для ефективного використання такого інформаційного багатства, окрім розвитку апаратного забезпечення, засобів зберігання та передачі даних, необхідно також мати зручні інтерфейси взаємодії між людиною і комп'ютером. Вони дають змогу користувачеві формулювати запити, переглядати файли, оновлювати або змінювати дані, а також приймати рішення на основі наявної інформації. Саме для цього були створені спеціалізовані програмні рішення – системи управління базами даних (СУБД).

Сучасні системи управління базами даних (СУБД) орієнтовані на багатокористувацьке середовище та розроблені для ефективного керування інформаційними масивами, з якими можуть одночасно працювати кілька користувачів. Вони надають гнучкі інструменти для зберігання, обробки та аналізу даних, що є критично важливим у різних сферах діяльності.

Основні можливості сучасних СУБД включають:

- Підтримку таблиць і взаємозв'язків між ними – завдяки цьому можна створювати складні структури даних та легко взаємодіяти з пов'язаними інформаційними блоками.
- Зручний користувацький інтерфейс, який дає змогу вводити, редагувати, шукати інформацію, а також виводити її у вигляді текстових або графічних звітів.
- Інтегровані засоби високорівневого програмування, що дозволяють розробникам створювати власні прикладні рішення на основі СУБД – від простих форм введення даних до складних автоматизованих систем.

Розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі дозволить ефективно збирати оперативну інформацію для обліку та подальшого аналізу. Тому тема розробленої бакалаврської роботи являється актуальною.

Мета роботи – розробити інформаційну систему для обліку та контролю абонементів в спортивно-оздоровчому комплексі, призначену для надання оперативної інформації для обліку і подальшого її аналізу та зниження обсягів паперового документообігу.

Об'єкт дослідження – спортивно-оздоровчий комплекс.

Предмет дослідження – інформаційна система для контролю абонементів в спортивно-оздоровчому комплексі.

Для досягнення поставленої мети необхідно виконати низку конкретних завдань:

- провести детальне вивчення та аналіз предметної області;
- виокремити основні об'єкти, які функціонують у межах системи, та визначити характер зв'язків між ними;
- розробити логічну модель бази даних;
- створити саму базу даних, включно з таблицями, ключами, збереженими процедурами та діаграмою структури;
- спроектувати інформаційну систему для контролю абонементів спортивно-оздоровчого комплексу. Основною метою цієї системи є оперативне інформування керівництва, підготовка даних для подальшого аналізу, а також зменшення обсягів паперової документації.

Запропонована система контролю має працювати з актуальними даними в режимі реального часу, забезпечуючи накопичення інформації, що у перспективі дає змогу аналізувати діяльність комплексу за будь-який часовий відрізок.

Одним із ключових елементів впроваджуваної системи є модуль оптимізації зберігання інформації. Його завдання – підвищити швидкість обробки даних за рахунок вибору найбільш ефективного способу їх збереження відповідно до типу інформації. Завдяки цьому значно зменшується час, необхідний для підготовки звітів, а також підвищується продуктивність інших підсистем, які працюють із цими даними.

1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Інформаційні системи

Інформаційна система являє собою комплекс взаємопов'язаних економічних задач, що характеризуються інтенсивним обміном інформацією між складовими елементами. Під задачею в цьому контексті розуміється процес обробки інформації, для якого чітко визначено вхідні та вихідні дані.

Структура функціональних підсистем інформаційної системи значною мірою залежить від специфіки економічної системи, до якої вона належить. Враховуються такі чинники, як галузева належність, форма власності, масштаб підприємства та характер його господарської діяльності.

Функціональні підсистеми інформаційної системи можуть формуватися на основі різних підходів:

- предметний – орієнтація на об'єкти управління;
- функціональний – за видами управлінських функцій;
- проблемний – згідно з розв'язуваними задачами;
- змішаний (предметно-функціональний) – поєднання кількох підходів.

Залежно від предметної орієнтації інформаційної системи у господарській діяльності промислових підприємств, зазвичай виділяють такі функціональні підсистеми:

- управління збутом готової продукції;
- управління виробничими процесами;
- управління матеріально-технічним забезпеченням;
- управління фінансовими ресурсами;
- управління персоналом.

У межах функціональних підсистем інформаційної системи задачі розглядаються на всіх рівнях управління – від оперативного до стратегічного, що

дозволяє забезпечити інтеграцію інформаційних потоків у вертикальному напрямі, тобто між рівнями ієрархії управління.

З метою реалізації основних управлінських функцій традиційно виділяють чотири ключові підсистеми [2]:

- планування – формування стратегій і тактичних рішень;
- оперативне управління – координація поточної діяльності;
- облік – фіксація та систематизація даних про господарські операції;
- аналіз – оцінка ефективності діяльності й прийняття управлінських рішень.

Ілюстрацією підходу до формування функціональних підсистем відповідно до управлінських функцій є багатокористувацький мережевий комплекс повної автоматизації «Галактика». Він реалізує повний цикл управління підприємством через чотири контури автоматизації:

- контур планування;
- контур оперативного управління;
- контур обліку та контролю;
- контур аналітичної обробки даних.

Проблемний принцип формування підсистем відображає потребу в гнучкому та оперативному прийнятті управлінських рішень, орієнтованих на вирішення конкретних завдань у межах системи підтримки прийняття рішень (СППР). Прикладом таких підсистем можуть бути: модулі бізнес-планування, системи управління проектами тощо. Залежно від реалізації, ці підсистеми можуть бути як локальними інформаційними системами, що отримують дані з корпоративної інформаційної системи (наприклад, бізнес-планування на базі ППП *Project-Expert*), так і спеціалізованими модулями всередині корпоративної інформаційної системи, наприклад, інформаційною системою керівника.

На практиці найчастіше застосовується *змішаний предметно-функціональний підхід*, який поєднує характеристику господарської діяльності з виконуваними управлінськими функціями. Така структура дозволяє максимально адаптувати інформаційну систему до реальних потреб підприємства, забезпечуючи

як предметну деталізацію, так і функціональну цілісність. Побудова структури інформаційної системи відповідно до цього підходу має враховувати:

- організаційну структуру підприємства;
- характер виконуваних функцій управління;
- специфіку галузі.

Функціональний принцип передбачає поділ системи на підсистеми за видами управлінських функцій:

- перспективне планування розвитку;
- техніко-економічне планування;
- бухгалтерський облік та аналіз господарської діяльності.

Предметний принцип орієнтований на управління конкретними ресурсами й видами діяльності. Прикладом таких підсистем є:

- технічна підготовка виробництва;
- управління основним та допоміжним виробництвом;
- контроль якості продукції;
- управління матеріально-технічним постачанням;
- збут і реалізація готової продукції;
- управління персоналом.

1.2 Класифікація та створення інформаційних систем

Інформаційні системи можна класифікувати за масштабами й функціональними можливостями на три основні класи:

- малі інформаційні системи;
- середні інформаційні системи;
- великі (корпоративні) інформаційні системи, що обслуговують потреби організацій регіонального або національного рівня.

До *малих інформаційних систем* зазвичай належать рішення, призначені для роботи в межах невеликих підприємств або окремих структурних підрозділів. Основними характеристиками таких систем є:

- короткий життєвий цикл – обмежений час актуальності або використання;
- масове застосування – орієнтація на широкий ринок без глибокої адаптації до конкретного користувача;
- доступна вартість – порівняно невисока ціна розробки та впровадження;
- обмежені аналітичні можливості – відсутність повноцінних інструментів для глибокої обробки даних;
- залежність від розробника – навіть незначні зміни часто вимагають втручання програмістів;
- використання СУБД середнього рівня, таких як Clarion, FoxPro, Clipper, Paradox, Access та інші;
- уніфіковане програмне середовище – обмеження у виборі апаратного та системного забезпечення;
- мінімальний рівень захисту даних – відсутність розвинених механізмів інформаційної безпеки.

Середні інформаційні системи вирізняються рядом особливостей, які суттєво відрізняють їх від малих систем. Основні ознаки цього класу включають:

- тривалий життєвий цикл, із можливістю подальшої еволюції в корпоративні рішення;
- вбудовані інструменти аналітичної обробки даних, що дозволяють здійснювати управлінський аналіз;
- наявність спеціалізованого персоналу, відповідального за адміністрування як апаратних засобів, так і програмного забезпечення;
- реалізовані механізми інформаційної безпеки, які забезпечують захист конфіденційної інформації;
- співпраця з розробниками ПЗ, що передбачає супровід і оновлення компонентів системи.

Корпоративні інформаційні системи (KIC) – це найбільший і найскладніший клас ІС, орієнтований на задоволення потреб великих організацій і холдингів. Характерними рисами KIC є:

- тривалий життєвий цикл системи, що часто перевищує термін служби окремих її компонентів;
- різноманітність апаратного забезпечення, яке постійно оновлюється в процесі експлуатації;
- використання різноманітного програмного забезпечення, як власного, так і стороннього;
- масштабність та складність завдань, що охоплюють різні рівні управління;
- інтеграція знань з багатьох предметних областей, які взаємодіють у межах єдиної системи;
- висока аналітична потужність, що дозволяє здійснювати прогнозування, оптимізацію та моделювання бізнес-процесів;
- територіальна розподіленість, тобто можливість функціонування системи в межах різних географічних локацій із централізованим або децентралізованим управлінням.

Сьогодні існує значна кількість досліджень, присвячених опису продуктів, методик та технологій, орієнтованих на розробку невеликих і середніх інформаційних систем. Натомість питання створення великих інформаційних систем, які об'єднують в собі численні локальні ІС, залишаються майже поза увагою науковців і практиків. Це часто призводить до того, що для побудови масштабних систем обирають інструменти, початково не призначені для таких завдань. Як наслідок, реалізовані проекти втрачають потенціал для подальшого розвитку.

У цьому контексті на перший план виходить необхідність формування чіткої концепції інтеграції розподілених інформаційних систем, особливо на рівні регіональних структур, із застосуванням розподілених об'єктно-орієнтованих технологій. Технологія створення інформаційних систем є сукупністю прикладних інженерних знань, які використовуються протягом усього життєвого циклу

програмного забезпечення. Вона також визначає організаційні принципи та підходи до управління процесами розробки, подаючи їх у вигляді послідовних або паралельних етапів, на кожному з яких створюються відповідні проміжні продукти.

Невід'ємною частиною будь-якої технології є інструментальні засоби, що застосовуються на різних стадіях розробки. Обрання конкретної технології безпосередньо залежить від архітектурних рішень майбутньої інформаційної системи. Розглядаючи інформаційну систему як комплекс взаємопов'язаних складових, їх можна класифікувати за такими рівнями [3]:

- апаратний рівень: включає комп'ютери, периферійне устаткування, мережеву та телекомунікаційну інфраструктуру;
- системний та системно-залежний рівень: охоплює операційні системи, мережеві протоколи тощо;
- рівень прикладного середовища: охоплює middleware-рішення (зокрема CORBA, DCE, Tuxedo), системи керування базами даних (DBMS), Intranet-технології, OLAP-сервіси та комунікаційні інтерфейси.

1.3 Дворівнева, трирівнева та розподілена однорангова архітектура інформаційної системи

У практиці розробки розподілених інформаційних систем найбільш широке використання отримали *дворівнева та трирівнева архітектури управління*.

Дворівнева архітектура передбачає наявність двох рівнів взаємодії. Один з компонентів виконує роль сервера, що надає певний набір сервісів, якими, у свою чергу, користується клієнтський компонент. Останній ініціює звернення до сервера з метою отримання необхідних послуг у ході виконання прикладних задач.

Фізичне розміщення цих компонентів може бути як на одному комп'ютері, так і на різних пристроях, з'єднаних через комп'ютерну мережу. Основна відмінність між реалізаціями дворівневої архітектури полягає у розподілі

функціональних обов'язків між клієнтом і сервером. Різні підходи до цього поділу обумовлюють варіативність моделей побудови ІС за дворівневою схемою.

У межах такої архітектури функціональні складові системи – інтерфейс користувача, прикладна логіка та система зберігання даних – можуть бути розміщені та розподілені між клієнтом і сервером у різних комбінаціях, що й утворює кілька варіантів її реалізації.

На основі існуючих підходів до декомпозиції функцій у дворівневій архітектурі інформаційних систем, можна виокремити такі її типи:

- архітектура з "інтелектуальним клієнтом", яка передбачає, що клієнтський компонент виконує значну частину прикладної логіки, а доступ до даних здійснюється через віддалений файловий сервер;
- архітектура з "інтелектуальним сервером", при якій клієнт діє здебільшого як термінал, а основна обробка даних та логіка реалізуються на стороні серверу баз даних;
- архітектура з розподіленою логікою, де функціональні обов'язки між клієнтом і сервером поділені більш гнучко, залежно від специфіки задач.

Попри відносну простоту реалізації, використання дворівневої архітектури для створення регіональних інформаційних систем вважається малоефективним. Це пояснюється низкою її структурних обмежень і технічних недоліків, які стають особливо відчутними при масштабуванні та інтеграції різнорідних інформаційних потоків на регіональному рівні.

Трирівнева архітектура вирізняється меншою жорсткістю зв'язків між клієнтськими та серверними частинами системи та забезпечує більш гнучкий підхід до організації розподіленої обробки даних. Найбільш поширеною є модель, у якій виділяються три логічно незалежні компоненти:

- рівень представлення (інтерфейс користувача),
- прикладний рівень (сервер додатків),
- рівень доступу до інформаційних ресурсів (база даних).

Ці компоненти функціонують автономно та взаємодіють між собою через стандартизовані інтерфейси з використанням засобів міжпроцесної комунікації [4].

Фізичне розміщення компонентів може варіюватися: вони можуть працювати як на одному комп'ютері, так і бути розподіленими між кількома машинами в мережі. Наприклад, інтерфейсний модуль зазвичай розташовується на персональному комп'ютері користувача, сервер додатків – на окремому сервері середнього рівня під управлінням систем, таких як Unix або Windows NT, а база даних – на потужному серверному обладнанні (Unix-сервери, мейнфрейми або міні-ЕОМ).

Важливою особливістю трирівневої архітектури є центральна роль сервера додатків. Саме тут зосереджено реалізацію основних прикладних функцій, кожна з яких представлена у вигляді окремого сервісу. Ці сервіси доступні всім клієнтам, які потребують відповідної функціональності. В одній системі може бути декілька серверів додатків, і кожен із них може надавати свій набір сервісів.

Ще однією перевагою цієї архітектури є високий рівень ізоляції між компонентами. Деталі реалізації прикладної логіки приховані від клієнтів, що дозволяє змінювати або переносити окремі частини системи без впливу на інші. Це значно спрощує супровід, модернізацію та масштабування інформаційної системи.

За оцінками фахівців у сфері інформаційних технологій, одним із найбільш перспективних напрямів еволюції сучасних інформаційних систем є *розподілена однорангова архітектура управління (peer-to-peer)*.

Особливістю цієї моделі є більш широке трактування ролі клієнта, що взаємодіє із сервером додатків. На відміну від традиційного підходу, клієнт у такій архітектурі вже не обмежується лише функцією представлення або інтерфейсом користувача. Він може реалізовувати прикладну логіку, а в окремих випадках – навіть виступати як сервер додатку.

У загальному вигляді будь-який компонент системи може бути водночас і постачальником, і споживачем сервісів. Тобто як клієнт, так і сервер, у цій архітектурі набувають властивостей взаємозамінності та рівноправності, формуючи однорангове середовище взаємодії.

Цей підхід відкриває можливість гнучкої декомпозиції функцій між елементами інформаційної системи. Під час проектування системи можна

здійснити розподіл функціональних завдань (інтерфейс, прикладна логіка, доступ до даних) таким чином, щоб це максимально відповідало специфіці задачі та умовам її реалізації. Завдяки цьому досягається вища адаптивність та масштабованість, що особливо важливо для систем, які повинні динамічно реагувати на зміни навантаження або конфігурації.

Для ефективної взаємодії компонентів інформаційної системи, яка функціонує на основі розподіленої однорангової архітектури, необхідно створити проміжний програмний шар (middleware). Саме він забезпечуватиме приймання запитів від клієнтів та їхнє перенаправлення на відповідні сервери. На щастя, сучасні розробники мають у своєму розпорядженні широкий вибір інструментів і технологій, які дають змогу будувати розподілені ІС без необхідності занурення у складнощі реалізації клієнт-серверної взаємодії. Значна частина таких рішень ґрунтується на стандарті CORBA (Common Object Request Broker Architecture), а деякі інструментальні пакети, як-от Orbix від компанії IONA Technologies, пропонують навіть розширені можливості цієї технології.

На основі проведеного аналізу можна дійти висновку, що розподілена однорангова архітектура є більш універсальною у порівнянні з дворівневими та трирівневими підходами. Завдяки чіткому поділу логічних компонентів і вдалому вибору програмних рішень, розподілені ІС досягають вищого рівня гнучкості, відкритості й продуктивності – рівня, який наразі залишається недосяжним для традиційних дворівневих і трирівневих архітектур.

Для ефективної взаємодії між компонентами інформаційної системи, що реалізує розподілену однорангову архітектуру, необхідно впровадити спеціальний проміжний програмний рівень – middleware. Саме цей рівень відповідає за приймання запитів від клієнтів та їх маршрутизацію до відповідних серверів або сервісів у системі.

На щастя, сьогодні розробникам доступний широкий спектр готових або анонсованих інструментальних засобів, які значно спрощують побудову розподілених ІС. Завдяки ним немає потреби заглиблюватися у складні технічні деталі реалізації взаємодії між клієнтами та серверами. Найпоширенішим серед

таких рішень є стандарт CORBA (Common Object Request Broker Architecture) – платформа, що забезпечує взаємодію об'єктів у розподіленому середовищі незалежно від їхнього фізичного розташування. Крім того, існують і розширені інструментальні пакети, наприклад, Orbix від компанії IONA Technologies, які пропонують додаткову функціональність поверх стандарту CORBA.

Підсумовуючи розгляд архітектур, можна дійти висновку, що розподілена однорангова архітектура має найбільш універсальний і гнучкий характер порівняно з дворівневою та трирівневою моделями. Чітке логічне розмежування функцій між компонентами, поєднане з раціональним використанням сучасних програмних платформ, дозволяє досягти високого рівня гнучкості, відкритості та продуктивності інформаційної системи – рівня, що в багатьох випадках недоступний за класичних архітектурних підходів.

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ СТВОРЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ УПРАВЛІННЯ ЦЕНТРОМ

2.1 Вибір бази даних для інформаційної системи

У сучасному світі інформаційних технологій бази даних відіграють надзвичайно важливу роль. Одним із ефективних інструментів для роботи з ними є *Firebird* – реляційна система керування базами даних, яка поєднує в собі легкість, швидкодію та відкритість.

Firebird працює на багатьох операційних системах – Windows, Linux, macOS, FreeBSD – і підтримує потужні механізми транзакцій, запитів, індексації та безпеки.

Firebird – це потужна SQL-сумісна система для зберігання, обробки та аналізу структурованих даних. Вона з'явилася у 2000 році як продовження відкритої версії InterBase від компанії Borland. Firebird може використовуватись як у вигляді серверної програми, так і у вбудованому режимі (embedded), що робить її універсальним рішенням для різних програмних застосунків.

Основні характеристики:

- Реляційна модель: Дані організовані у вигляді таблиць, пов'язаних між собою через ключі.
- Мова SQL: Firebird підтримує стандартну мову запитів SQL, включаючи транзакції, процедури, тригери та представлення.
- Zero maintenance: Firebird не потребує постійного адміністрування – це робить її чудовим варіантом для вбудованих систем або невеликих компаній.
- Вбудована або серверна архітектура: Працює як локально (embedded), так і у вигляді сервера (client-server).

Основні переваги:

- Безкоштовність і відкритий код. Firebird повністю безкоштовна для будь-якого використання – навіть комерційного, що зменшує витрати на розробку.

Firebird не має жодних прихованих витрат чи обмежень.

- Простота встановлення та використання. Інсталяція Firebird займає кілька хвилин. Вбудована версія (embedded) взагалі не потребує сервера – ви просто підключаєте .fdb файл, як базу даних.
- Механізми безпеки. Підтримка прав доступу, захист паролем, ролі користувачів – усе це дозволяє надійно обмежувати доступ до інформації.
- Підтримка складної логіки. Firebird підтримує збережені процедури, тригери, генератори – що дозволяє переносити частину бізнес-логіки з програми безпосередньо в базу.
- Кросплатформність. Система працює на різних операційних системах: Windows, Linux, macOS.
- Мінімальне адміністрування. Firebird не вимагає постійного технічного обслуговування, що зручно для невеликих організацій.
- Підтримка SQL. Повноцінна підтримка мови запитів SQL, включно з транзакціями, процедурами, тригерами, що забезпечує гнучкість у створенні логіки додатків.
- Висока продуктивність. Firebird забезпечує швидку обробку даних навіть при великих обсягах, завдяки оптимізованому рушію. Це дозволяє обслуговувати десятки тисяч записів без втрати продуктивності.

Порівняння з іншими РСКБД надається в таблиці 2.1.

Таблиця 2.1 – Порівняння Firebird з іншими РСКБД

Система	Ліцензія	Розмір	Продуктивність	Потреба в адмініструванні
Firebird	Безкоштовна	Легка	Висока	Мінімальна
MySQL	Open Source / Commercial	Середня	Висока	Середня
PostgreSQL	Open Source	Важча	Висока	Гнучка, потребує налаштування
SQLite	Public Domain	Дуже легка	Середня	Мінімальна

Firebird ідеально балансує між функціональністю і простотою, що робить її зручною як для новачків, так і для досвідчених розробників.

Firebird широко застосовується в багатьох галузях, включаючи:

- Торгівлю (касові апарати, облік товару).
- Банківське програмне забезпечення.
- Медичні системи.
- Управління бізнесом (ERP).
- Освітні проєкти (електронні журнали, системи тестування).

Тобто завдяки легкості, високій продуктивності та надійності Firebird широко застосовується у фінансових, торгових, медичних та освітніх системах; ідеально підходить для створення бізнес-програм, облікових систем, систем управління персоналом та навчального процесу.

Firebird – це сучасна, надійна та зручна система управління базами даних, яка має безліч переваг. Її легкість у використанні, доступність і гнучкість роблять її чудовим вибором як для початківців, так і для досвідчених розробників. У світі, де дані мають ключове значення, Firebird залишається перевіреним інструментом для їх ефективного зберігання та обробки.

2.2 Вибір мови програмування клієнтської частини

Сучасний світ інформаційних технологій неможливо уявити без інтернет-ресурсів. Веб-сайти стали невід'ємною частиною нашого життя. Веб-сайт є важливим засобом для наочного подання інформації. Його наповнення – тексти, зображення, аудіо- та відеофайли – виконує не лише інформаційну, а й естетичну функцію. Вигляд сайту формується завдяки вдало підібраним елементам дизайну, композиції та інтерфейсу користувача. Продумане оформлення забезпечує легкість сприйняття контенту, сприяє зручній навігації та утримує інтерес відвідувачів. Натомість недоладний дизайн може ускладнити доступ до потрібної інформації або зменшити її цінність в очах користувача.

Якісне створення та розвиток сайтів вимагають знань низки технологій, серед яких HTML, CSS та JavaScript займають центральне місце. Саме взаємодія цих трьох компонентів забезпечує створення сучасних, інтерактивних та естетично привабливих сторінок: HTML визначає структуру сторінки, CSS відповідає за її стильове оформлення, а JavaScript надає сайту інтерактивності й динаміки. Їхнє вдале поєднання дозволяє створювати сучасні, зручні та привабливі веб-додатки, орієнтовані на потреби користувачів.

2.2.1 HTML (Hypertext Markup Language)

HTML (від англ. *HyperText Markup Language* – «мова гіпертекстової розмітки») – це стандартизована мова для створення гіпертекстових документів, призначених для перегляду у веб-браузерах. Браузери отримують HTML-документи з веб-серверів за допомогою протоколів HTTP або HTTPS, або ж відкривають їх із локального носія. Після отримання файлу браузер інтерпретує HTML-код і перетворює його на графічний інтерфейс, який відображається на екрані користувача.

HTML є невід'ємної складової веб-розробки, дозволяючи створювати структуру веб-сторінок. Він визначає розташування тексту, зображень, відео, посилань і інших елементів на сторінці за допомогою тегів, які надають контенту специфічну семантику. HTML-теги дозволяють точно визначати, де розташований кожний елемент на сторінці, а також установлюють їхньої властивості й зв'язку між ними.

- Створення структури веб-сторінок.
- Визначення розташування тексту, зображень, відео, посилань і інших елементів.
- Використання тегів для додання контенту специфічної семантики.

HTML-теги є основними будівельними блоками веб-розробки, використовуваними для визначення структури й семантики веб-сторінок. Кожний

тег має свою унікальну функцію й визначає роль елемента на сторінці. Теги можуть бути парними (з відкриваючим і закриваючим тегом) або непарними (без закриваючого тегу). Деякі з найбільш використовуваних HTML-тегів включають:

- **div**: Цей тег використовується для угруповання та розподілу різних частин сторінки. Він часто використовується для створення блоків з однорідним змістом, які можна стилізувати за допомогою CSS.
- **p**: Тег для визначення абзацу тексту. Увесь текст між відкриваючим і закриваючим тегом розглядається як один абзац.
- **a**: Використовується для створення посилань. Атрибут `href` указує адресу, на яку слід перейти після натискання на посилання.
- **img**: Цей тег вставляє зображення на сторінку. Атрибут `src` указує шлях до файлу зображення.
- **ol** та **li**: Використовуються для створення нумерованого списку. Тег **ol** визначає сам список, а кожний пункт списку відображається за допомогою тегу **li**.
- **ul** та **li**: Використовуються для створення маркованого списку. Тег **ul** визначає сам список, а кожний пункт списку відображається за допомогою тегу **li**.
- **table**: Використовується для створення таблиць на сторінці. Теги визначають рядки таблиці, а або комірки в цих рядках.

Використання різних HTML-тегів дозволяє розміщати й відображати контент на веб-сторінках відповідно до їхнього призначення й семантикою.

Елементи HTML є основними будівельними блоками веб-сторінок. Завдяки HTML можна вбудовувати на сторінку різноманітні структури, зображення та інші об'єкти, зокрема інтерактивні веб-форми. Мова HTML надає засоби для створення заголовків, абзаців, списків, посилань, цитат та інших елементів оформлення контенту. Елементи HTML позначаються тегами, які записуються у кутових дужках. Деякі теги, наприклад `<img>` або `<input>`, безпосередньо вбудовують контент на сторінку. Інші, як-от `<p>`, охоплюють текстовий вміст та можуть містити в собі інші теги як піделементи.

Браузери самі теги не відображають – вони використовують їх для побудови структури та оформлення сторінки.

2.2.2 CSS (Cascading Style Sheets)

CSS (англ. *CascadingStyleSheets* «каскадні таблиці стилів») – формальна мова декорування й опису зовнішнього вигляду документа (веб-сторінки), написаного з використанням мови розмітки (найчастіше HTML або XHTML). Також може застосовуватися до будь-яких Xml-XML-документів, наприклад, до SVG або XUL.

До впровадження CSS оформлення веб-сторінок здійснювалося виключно засобами HTML і прописувалося безпосередньо у структурі документа. З появою CSS стало можливим принципово розділити зміст і візуальне представлення сторінки. Завдяки цьому нововведенню розробники отримали змогу легко застосовувати єдині стилі оформлення до великої кількості подібних документів, а також оперативно змінювати вигляд веб-ресурсу без необхідності редагувати кожену сторінку окремо.

CSS (Каскадні таблиці стилів) є невід'ємної складової веб-розробки, відповідальної за стильове оформлення веб-сторінок з метою зробити їхніми привабливими й естетично приємними для користувачів. Він дозволяє налаштовувати різноманітні параметри, що впливають на вид і відображення елементів на сторінці. Розшифрування "каскадні" у назві CSS вказує на те, що стилі можуть бути застосовані до елементів веб-сторінки в різних місцях і різним способом, при цьому застосовані стилі можуть перекриватися або успадковуватися від батьківських елементів.

Для досягнення стилізації веб-сторінок CSS надає ряд зручних *можливостей*:

- Кольору: Визначення кольору тексту, тла, рамок і інших елементів на сторінці.
- Шрифти: Установка типу шрифту, розміру шрифту, жирності, нахилу й

інших параметрів тексту.

- Розміри: Контроль над розмірами елементів, такими як ширина, висота, відступи, поля й інші розмірні властивості.
- Відступи й границі: Визначення відстаней між елементами, відступів від країв і границь елементів.
- Селектори: CSS використовує селектори для вибору конкретних елементів сторінки, до яких будуть застосовуватися стилі. Це може бути вибірка по класах, ідентифікаторах, тегах і іншим атрибутам.

Інтеграція CSS у веб-розробку дозволяє створювати стильні й привабливі веб-сторінки, які коректно відображаються на різних пристроях і розмірах екранів, що є ключовим аспектом сучасного веб-дизайну.

CSS активно використовується творцями веб-сторінок для визначення кольорів, шрифтів, стилів, розташування елементів та інших аспектів візуального оформлення веб-ресурсів. Головна мета впровадження CSS полягає у чіткому розмежуванні логічної структури веб-сторінки (яку описують засобами HTML чи інших мов розмітки) та її зовнішнього вигляду (який задається мовою CSS). Такий поділ сприяє підвищенню доступності документів, забезпечує більшу гнучкість у керуванні їхнім оформленням і зменшує складність та дублювання у структурному коді.

Крім того, CSS відкриває можливість представлення одного й того самого документа у різних форматах: для екранного відображення, для друку, для озвучення за допомогою голосових браузерів чи програм зчитування екрана, а також для пристроїв, що працюють із шрифтом Брайля.

- *Переваги:*
- Адаптивність дизайну для різних пристроїв. Дизайн сторінки підлаштовується під розміри екрана: на великому моніторі розміщення елементів оптимізоване для широкого відображення, при друку меню автоматично приховується, а на КПК чи смартфонах меню переміщується під основний вміст для зручності користувача.
- Оптимізація швидкості завантаження сайту шляхом винесення стилів у

окремий CSS-файл. У такому випадку браузер завантажує лише структуру документа та контент сторінки, а стилі підтягує окремо й кешує, що значно скорочує час повторних завантажень.

- Легкість у зміні зовнішнього вигляду. Для оновлення дизайну більше не потрібно редагувати кожен HTML-сторінку – достатньо внести зміни лише в один CSS-файл, що істотно спрощує обслуговування сайту.
- Розширені можливості оформлення. Завдяки CSS можна створювати блоки, навколо яких текст акуратно обтікає (наприклад, навігаційне меню), або зробити так, щоб меню залишалося завжди видимим при прокрутці сторінки, підвищуючи зручність навігації.

Недоліки:

- Відмінності у відображенні верстки в різних браузерах. Особливо це стосується застарілих версій браузерів, які можуть по-різному інтерпретувати одні й ті самі CSS-правила, що призводить до непередбачуваного вигляду сторінок.
- Потреба змінювати не лише CSS, а й HTML. На практиці нерідко доводиться коригувати не тільки стилі у CSS-файлах, а й самі HTML-теги, які складно і заплутано пов'язані із CSS-селекторами. Це ускладнює редагування та тестування і може звести нанівець переваги централізованого керування стилями.

Розташування та формат правил CSS.

Правила оформлення можуть бути як вбудовані безпосередньо у веб-документ, так і винесені в окремі файли з розширенням .css. CSS-файл – це звичайний текстовий документ, що містить набір правил форматування та коментарі до них.

Способи підключення стилів до веб-сторінки.

Існує чотири основні методи інтеграції CSS: безпосередньо у тегах HTML, у заголовку документа (<style>), шляхом імпорту зовнішнього файлу, або за допомогою зовнішнього посилання (<link>).

Правила CSS можуть розташовуватися як у самому веб-документі, зовнішній вигляд якого вони описують, так і в зовнішніх файлах, що мають розширення .css. Формат CSS – це текстовий файл, в якому міститься перелік правил CSS і коментарів до них. Стилї CSS можуть бути підключені або впроваджені в описуваний ними веб-документ чотирма способами:

- Підключення зовнішнього файлу стилів за допомогою елемента `<link>`. Якщо CSS-опис зберігається в окремому файлі, його можна підключити до веб-документа через тег `<link>`, розміщений у секції `<head>`.
- Імпорт файлу стилів через директиву `@import`. Коли файл стилів зберігається окремо, його можна підключити також за допомогою інструкції `@import` всередині блоку `<style>`, який розташовується у секції `<head>`.
- Вбудовані стилі в документі. Стилї можуть бути описані безпосередньо у тілі документа в елементі `<style>`, який також розміщується у `<head>`.
- Використання атрибутів для локального форматування. Стилї можна задавати окремим елементам безпосередньо через атрибут `style` всередині тегу.

У перших двох випадках ми працюємо з *зовнішніми стилями*, які фізично знаходяться поза межами HTML-документа. У двох останніх – зі *внутрішніми стилями*, які прописуються безпосередньо у самому документі.

Підключення CSS до XML-документів. Для використання CSS із XML-документами потрібно додати в них спеціальне посилання на файл стилів, що дозволяє правильно застосувати правила оформлення:

```
<?xml-stylesheet type="text/css" href="style.css"?>
```

Застосування CSS у HTML-документах базується на двох основних принципах – *спадкування* та *каскадування*.

Спадкування. Властивості, задані для батьківських елементів, автоматично передаються їхнім нащадкам. Це означає, що елементи всередині іншого елемента успадковують певні стилі, що спрощує та уніфікує оформлення сторінки.

Каскадування. Цей принцип вступає в дію, коли до одного HTML-елемента застосовується кілька CSS-правил одночасно, і між ними виникає конфлікт. Для

врегулювання таких ситуацій існує система пріоритетів, яка визначає, яке саме правило має переважати:

- Найнижчий пріоритет мають стилі, встановлені браузером за замовчуванням.
- Вищий пріоритет отримують стилі, задані користувачем у налаштуваннях свого браузера.
- Найвищий пріоритет належить стилям, визначеним автором веб-сторінки. У межах авторських стилів пріоритети розподіляються так:
 - Наслідувані стилі (успадковані від батьківських елементів) мають найнижчий пріоритет.
 - Стилi з зовнішніх CSS-файлів, підключених до документа, мають вищий пріоритет.
 - Стилi, прописані безпосередньо в `<style>` контейнерах документа за допомогою різних типів селекторів, мають ще більший пріоритет.

2.2.3 JavaScript

Javascript – це високорівнева скриптова мова програмування, яка широко використовується для створення інтерактивних додатків і сайтів. Javascript є важливої складової веб-розробки, тому що він додає інтерактивність веб-сторінкам, роблячи їх більш привабливими й функціональними для користувачів. Ця мова програмування дозволяє створювати різноманітні функції, які можуть взаємодіяти з відвідувачами сторінки в реальному часі.

Можливості Javascript:

- Обробка подій: Javascript може реагувати на різні події, такі як натискання миші, введення даних користувача або завантаження сторінки. Наприклад, він може виконувати певні дії при натисканні на кнопку або введенні даних у форму.
- Маніпуляція DOM: Javascript дозволяє змінювати вміст і структуру веб-сторінок у реальному часі, використовуючи DOM (Document Object Model).

Наприклад, він може додавати, видаляти або змінювати елементи сторінки залежно від дій користувача.

- Валидація форм: Javascript може перевіряти дані, введені користувачем у форму, перед їх відправленням на сервер. Це дозволяє запобігти некоректному введенню даних і забезпечити безпеку веб-додатків.
- Анімація й візуальні ефекти: За допомогою Javascript можна створювати різні анімаційні ефекти, такі як зміна розміру, переміщення або зміна кольору елементів. Це дозволяє створювати привабливі й динамічні веб-сторінки.
- Взаємодія із сервером: Javascript може взаємодіяти із сервером для одержання або відправлення даних без перезавантаження сторінки. Це відкриває можливості для реалізації асинхронного завантаження контенту й створення динамічних веб-додатків.

Таким чином, Javascript є потужним інструментом, який дозволяє створювати різноманітні й інноваційні веб-додатки, поліпшуючи взаємодію користувачів з веб-сайтами.

Тобто JavaScript – це те, що вдихає життя в статичні веб-сторінки: додає анімацію, спливаючі повідомлення, перевіряє форми на правильність заповнення, оновлює контент без перезавантаження і робить інтерфейс "живим" та чуйним. Наприклад: наводите курсор на кнопку – і вона змінює колір або зникає після кліку. Завдяки JavaScript сторінки стають справжніми інтерактивними середовищами.

Основні характеристики JavaScript:

- Інтерпретована мова. JavaScript виконується безпосередньо в браузері, без попередньої компіляції. Розробник змінює код – і миттєво бачить результат, що значно пришвидшує розробку.
- Тісна інтеграція з HTML та CSS. У поєднанні з ними JavaScript дозволяє створювати повноцінні динамічні сайти.
- Динамічна типізація. Типи змінних визначаються автоматично під час виконання коду, що робить процес програмування гнучкішим і швидшим.

- Багатство бібліотек і фреймворків. Сотні готових рішень (наприклад, React, Vue.js, jQuery) суттєво скорочують час розробки.
- Кросплатформеність. Один і той самий код працює на різних пристроях і операційних системах без змін.
- Автоматичне управління пам'яттю. Невикористовувані об'єкти прибираються автоматично, що полегшує оптимізацію програм.
- Підтримка різних парадигм програмування. JavaScript дозволяє писати імперативний, об'єктно-орієнтований і функціональний код, забезпечуючи гнучкість у вирішенні будь-яких завдань.
- Асинхронність. Механізми, як-от promises і async/await, допомагають обробляти довгі операції (наприклад, запити до сервера) без "заморожування" сторінки.
- Робота з DOM. Вбудована підтримка моделі документа дозволяє легко змінювати вміст, реагувати на події та створювати динамічні інтерфейси.
- Широке співтовариство. Мільйони розробників по всьому світу щодня вдосконалюють екосистему JavaScript: створюють нові бібліотеки, пишуть підручники, проводять конференції – тому завжди є де знайти підтримку.
- Відкритий код. JavaScript є безкоштовним і відкритим – як Python, він не має ліцензійних обмежень і доступний для всіх бажаючих.

JavaScript – це справжній двигун сучасної Frontend-розробки й не тільки. У тандемі з HTML і CSS він допомагає створювати яскраві, інтерактивні та зручні сайти.

Переваги JavaScript:

- Легкість у вивченні. Зрозумілий, логічний синтаксис робить JavaScript чудовим вибором для початківців, які лише починають занурюватися у світ програмування.
- Універсальність. JavaScript працює практично на будь-якому пристрої та платформі. Все, що потрібно для старту – звичайний браузер.

- Гнучкість і сумісність. Мова легко інтегрується з HTML, CSS та іншими технологіями, підтримує різні стилі програмування, працює з асинхронністю й динамічною типізацією.
- Широкий арсенал інструментів. Бібліотеки та фреймворки на JavaScript (наприклад, React, Angular, Node.js) дозволяють вирішувати найрізноманітніші завдання: від створення адаптивних сайтів до розробки складних серверних додатків.
- Світова популярність. JavaScript є основою багатьох відомих компаній, таких як Netflix, Amazon, PayPal, Facebook, Twitter, Google, Airbnb та інших. Вивчивши цю мову, ви значно підвищите свої шанси знайти цікаву та добре оплачувану роботу.

Недоліки Javascript:

- Нижча продуктивність у порівнянні з іншими мовами. У задачах, що потребують інтенсивних обчислень або обробки великих обсягів даних, JavaScript може поступатися за швидкістю спеціалізованим мовам програмування, таким як C++ або Java.
- Уразливість до кібератак. Наприклад, JavaScript може стати мішенню для атак типу XSS (міжсайтовий скриптинг). Водночас варто пам'ятати: проблеми безпеки – це не лише проблема JavaScript, а загальна загроза для багатьох технологій.
- Проблеми з масштабуванням у великих проєктах. Динамічна типізація, яка є зручною для швидкої розробки, у великих системах може ускладнювати підтримку коду та збільшувати витрати на його обслуговування.
- Несумісність між браузерами. Хоча сучасні браузери дедалі краще підтримують JavaScript, все ще трапляються ситуації, коли один і той самий код поводить по-різному у різних середовищах.

JavaScript давно вийшов за межі простої взаємодії з веб-сторінками та перетворився на справжній універсальний інструмент. Сьогодні JavaScript застосовують у різних сферах:

- Фронтенд-розробка. Створення інтерактивних елементів – слайдерів, спливаючих вікон, форм зворотного зв'язку, анімацій і обробки подій. JavaScript робить інтерфейси зручними, привабливими та динамічними для користувача.
- Бекенд та фулстек. Завдяки платформі Node.js JavaScript використовується на серверній стороні – для створення API, портативних застосунків і сервісів, що безперервно обмінюються даними з користувачами.
- Односторінкові застосунки (SPA). Бібліотеки та фреймворки на кшталт React, Vue.js і Angular допомагають будувати швидкі, інтерактивні додатки з плавною роботою без постійного перезавантаження сторінок (приклади – Airbnb, Gmail, Pinterest, Netflix).
- Мобільна розробка. Кросплатформені мобільні додатки на базі React Native, Ionic або NativeScript дозволяють створювати повноцінні застосунки для Android і iOS за допомогою знань JavaScript.
- Ігрова індустрія. Легкі браузерні ігри – пригодницькі бродилки, головоломки, гонки, стратегії – все це можна реалізувати на JavaScript.
- Інтернет речей (IoT). Створення систем для розумних будинків, збирання даних із пристроїв та керування ними також можливе за допомогою JavaScript.
- Машинне навчання. Хоча Python домінує у сфері AI та ML, JavaScript через бібліотеки типу TensorFlow.js дозволяє інтегрувати моделі машинного навчання безпосередньо у веб-додатки.
- Розширення для браузерів. Від управління паролями й створення скріншотів до VPN та блокування реклами – всі ці можливості відкриває JavaScript через розробку розширень.

Отже, сьогодні JavaScript – це цілий світ можливостей для різноманітних проєктів і сфер застосування.

2.2.4 Взаємодія між HTML, CSS та JavaScript

Взаємодія між HTML, CSS і Javascript є ключовим для створення повноцінного веб-сайту, який поєднує в собі структуру, зовнішній вигляд і функціональність. HTML виступає як основа, визначаючи кістяк сторінки і її елементи, такі як заголовки, абзаци, зображення й посилання. CSS відповідає за стилізацію цих елементів, роблячи їх зовнішній вигляд більш привабливим і організованим. За допомогою CSS можна змінювати кольори, розміри, шрифти й інші властивості, щоб досягтися бажаного виду сторінки.

Javascript вносить важливий елемент інтерактивності. Javascript дозволяє створювати складні функції й оброблювачі подій, які реагують на дії користувача. Зокрема, Javascript використовує DOM (Document Object Model), щоб взаємодіяти з HTML-елементами. Це означає, що Javascript може змінювати вміст і структуру сторінки в реальному часі, забезпечуючи динамічну поведінку. Наприклад, при натисканні на кнопку Javascript може додавати новий елемент на сторінку або змінювати властивості існуючих елементів.

Ця взаємодія стає ще могутніше, коли Javascript і CSS працюють разом. Javascript може додавати або змінювати класи HTML-елементів, що дозволяє застосовувати нові стилі або змінювати існуючі залежно від дії користувача. Наприклад, при наведенні курсору на елемент, Javascript може додати клас, який змінить його колір або розмір. Таким чином, спільна робота HTML, CSS і Javascript дозволяє створювати багатофункціональні й динамічні веб-сайти, які вражають своїм зовнішнім виглядом і функціональністю.

HTML, CSS і Javascript - це важливі інструменти для сучасної веб-розробки, їх інтеграція дозволяє створювати гарні, інтерактивні й функціональні веб-сайти. Оволодіння цими мовами і їх взаємодією є ключовим етапом для будь-якого веб-розроблювача.

Вони не тільки допомагають забезпечити ефективне структурування, оформлення й взаємодія веб-сторінок, але й відкривають безліч можливостей для творчості й інновацій. Компетентне володіння HTML, CSS і Javascript дозволяє

створювати вражаючі веб-додатки, які відповідають високим стандартам якості та відповідають потребам користувачів у сучасному цифровому середовищі.

2.3 Організація інтеграції клієнтської та серверної частин

Сучасні веб-застосунки складаються з двох основних частин:

- *Клієнтська частина (frontend)* – те, що бачить і з чим взаємодіє користувач (HTML, CSS, JavaScript);
- *Серверна частина (backend)* – логіка, яка працює "за лаштунками": обробка запитів, доступ до баз даних, авторизація тощо.

Щоб ці частини "спілкувалися", використовують технології передачі даних, зокрема *AJAX*. У ролі сервера часто виступає *Node.js* – сучасне середовище виконання JavaScript, яке дозволяє створювати високопродуктивні сервери на тій же мові, що й фронтенд.

Node.js – це середовище, яке дозволяє запускати JavaScript на сервері, поза браузером. Воно базується на рушії V8 (той самий, що використовується в Google Chrome), і відзначається:

- Асинхронною, подієвою моделлю (event-driven).
- Підтримкою тисяч одночасних підключень.
- Роботою з мережею, файлами, базами даних.

Для створення сервера часто використовують *фреймворк Express.js* – мінімалістичний, але потужний інструмент для створення API.

AJAX (Asynchronous JavaScript and XML) – це технологія, яка дозволяє надсилати запити на сервер без перезавантаження сторінки. Хоча назва містить XML, сьогодні замість нього частіше використовують JSON.

Використовуючи AJAX (через *fetch*, *XMLHttpRequest* або бібліотеки типу *jQuery*), можна динамічно оновлювати контент сторінки, наприклад:

- Завантажувати повідомлення в чаті.
- Надсилати форми.

- Отримувати оновлення з бази даних.

Клієнт надсилає дані форми через AJAX. Сервер їх приймає, обробляє та повертає відповідь. Клієнт отримує повідомлення, не оновлюючи сторінку.

Переваги інтеграції через Node.js та AJAX

- Одна мова – JavaScript. І фронтенд, і бекенд можна писати на одній мові, що спрощує командну роботу та навчання.
- Швидкість та асинхронність. Node.js обробляє тисячі запитів без блокування. AJAX дозволяє працювати з сервером у фоновому режимі.
- Миттєвий обмін даними. Користувач бачить зміни одразу, що значно покращує взаємодію з сайтом.
- Зручна робота з API. AJAX легко працює з REST API, які часто реалізуються на Node.js.

Інтеграція клієнтської та серверної частин за допомогою Node.js і AJAX – це основа сучасної веброзробки. Такий підхід дозволяє створювати інтерактивні, швидкі та гнучкі застосунки, де користувач отримує дані миттєво, без оновлення сторінки. Вивчення цих технологій відкриває двері до створення повноцінних вебпрограм – від простих форм до складних SPA-додатків.

3 ОПИС ІНФОРМАЦІЙНОЇ СИСТЕМИ

3.1 Структура меню інформаційної системи

Структуру меню, яке складається з форм нормативно-довідкової інформації та вхідних і вихідних екрани форм, зображено на рисунку 3.1.



Рисунок 3.1 – Структура меню інформаційної системи

3.2 Опис системи контролю абонементів спортивно-оздоровчого комплексу

Система обліку абонементів спортивно-оздоровчого комплексу розроблена з метою автоматизації ключових процесів у роботі спортивного закладу. Вона суттєво полегшує документообіг та контроль за відвідуваністю клієнтів.

Інформаційна система дозволяє швидко додавати нових користувачів, реєструвати їх участь у спортивних секціях та формувати відповідні абонементи. Кожен візит клієнта фіксується, що дає змогу вести точну статистику відвідувань. Завдяки функціям пошуку та сортування можна легко знайти потрібну інформацію про будь-якого відвідувача. Інтерфейс системи інтуїтивно зрозумілий, а процес

введення та виведення даних – максимально зручний.

Довідник клієнтів надає можливість переглядати актуальний список користувачів комплексу, а також швидко додавати нових. Для кожного клієнта зберігаються такі дані: повне ім'я, серія та номер паспорта, номер телефону й електронна пошта. Передбачено гнучке сортування та пошук за всіма полями.

Довідник тренерів містить інформацію про викладацький склад, включно з ПІБ, контактними даними, спортивним званням і досягненнями.

Довідник спортивних секцій дозволяє вести реєстр усіх наявних секцій, вказуючи їх назви, графік занять, місячну вартість (у гривнях), а також імена відповідальних тренерів.

Абонементи – це розділ, що охоплює усі активні абонементи клієнтів, забезпечує ведення обліку відвідувань, створення та друк нових карток. Також реалізовано функціонал для пошуку та сортування інформації в таблиці.

3.3 Логічна структура бази даних

3.3.1 Нормативно-довідникова інформація

Нормативно-довідникова інформація надана в таблицях 3.1-3.3.

Таблиця 3.1 – Довідник тренерів (tr)

Код тренера	ПІБ	Телефон	E-mail	Звання	Досягнення
Kod_tr	Fio tr	Tel_tr	Email tr	Zvan_tr	Dost_tr
I4	C50	C10	C20	C250	C250

Таблиця 3.2 – Довідник спортивних секцій (spsec)

Код спортивної секції	Назва секції	Дні відвідувань	Вартість занять на місяць, грн.	Код тренера
Kod_spsec	Name_spsec	Dni	Stoim	Kod_tr
I4	C100	C50	N(4,2)	I4

Таблиця 3.3 – Довідник клієнтів (kl)

Код клієнта	ПІБ	Серія паспорта	Номер паспорта	Телефон	E-mail
Kod_kl	Fio kl	Serpas_kl	Nompas_kl	Tel_kl	Email kl
I4	C50	C2	I6	C10	C20

3.3.2 Вхідна інформація

Вхідну інформацію надано в таблиці 3.4.

Таблиця 3.4 – Абонементи (abon)

Номер абонементу	Код клієнта	Код спортивної секції	Дата покупки	Відвідування 1	Відвідування 2
Kod_abon	Kod_kl	Kod_spsec	Data_pok	Vid_1	Vid_2
I4	I4	I4	D8	boolean	boolean

Продовження таблиці 3.4 – Абонементи (abon)

Відвідування 3	Відвідування 4	Відвідування 5	Відвідування 6	Відвідування 7
Vid_3	Vid_4	Vid_5	Vid_6	Vid_7
boolean	boolean	boolean	boolean	boolean

Продовження таблиці 3.4 – Абонементи (abon)

Відвідування 8	Відвідування 9	Відвідування 10	Відвідування 11	Відвідування 12
Vid_8	Vid_9	Vid_10	Vid_11	Vid_12
boolean	boolean	boolean	boolean	boolean

3.4 Створення бази даних

На рисунку 3.2 показано створення бази даних Gym.gdb.

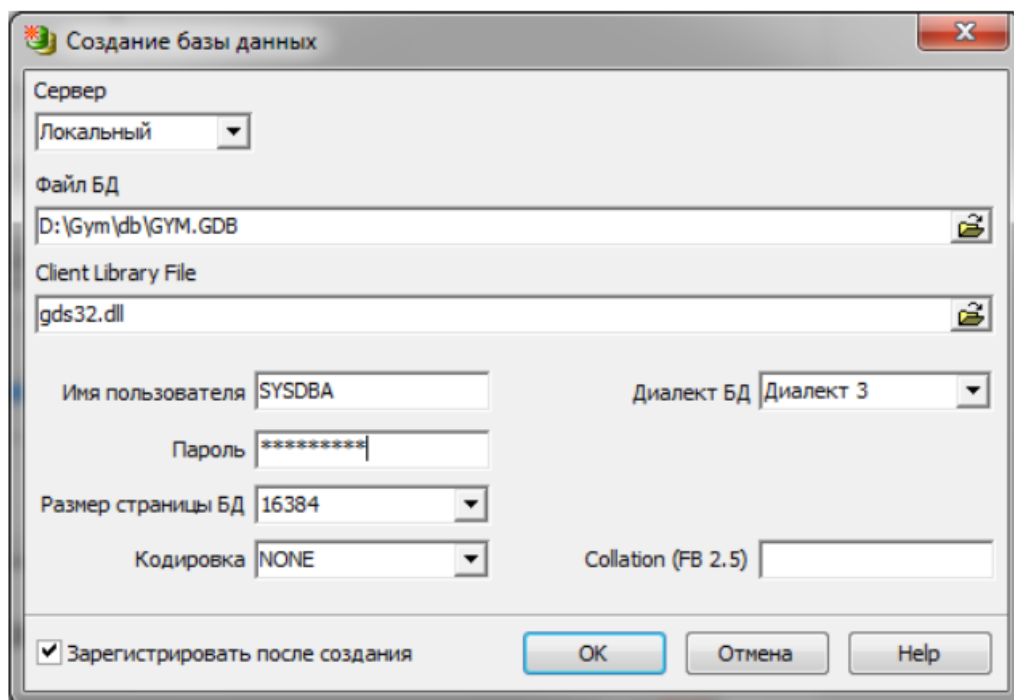


Рисунок 3.2 – Створення бази даних Gym.gdb

3.4.1 Створення таблиць бази даних

Створення таблиць бази даних Gym.gdb показано на рисунках 3.3-3.6.

#	PK	BK	UNQ	Название	Тип	Дочен	Длина	Точ...	Подтип	Масив	Не пу...	Кодировка	Коллате	Описан...	Вычислить как...	По умолчанию
1				KOD_KL	INTEGER						☒	NONE	NONE			
2				FTO_KL	VARCHAR		50				☐	NONE	NONE			
3				SERPAS_KL	VARCHAR		2				☐	NONE	NONE			
4				NOMPAS_KL	INTEGER						☐	NONE	NONE			
5				TEL_KL	VARCHAR		10				☐	NONE	NONE			
6				EMAIL_KL	VARC-AR		20				☐	NONE	NONE			

Рисунок 3.3 – Довідник клієнтів (kl)

Таблица : [TR] : D:\Gym\db\GYM.GDB (D:\Gym\db\GYM.GDB)

Количество записей: 1 Type: Persistor TR

Поля: Ограничения Индексы Зависимости Триггеры Данные Master/Detail View Описание Скрипт Права Протокол Сравнение Задачи

KOD_TR INTEGER NOT NULL

#	PK	BK	UNQ	Название	Тип	Дополн	Длина	Точ...	Подтип	Масштаб	Не ну...	Кодировка	Коллате	Описан...	Вычислить как...	По умолчанию
1				KOD_TR	INTEGER						☒					
2				FIO_TR	VARCHAR		50				☐	NONE	NONE			
3				TEL_TR	VARCHAR		10				☐	NONE	NONE			
4				EMAIL_TR	VARCHAR		20				☐	NONE	NONE			
5				ZVAN_TR	VARCHAR		250				☐	NONE	NONE			
6				DOST_TR	VARCHAR		250				☐	NONE	NONE			

Описание поля Объекты, использующие поле

Рисунок 3.4 – Довідник тренерів (tr)

Таблица : [SPSEC] : D:\Gym\db\GYM.GDB (D:\Gym\db\GYM.GDB)

Количество записей: 1 Type: Persistor SPSEC

Поля: Ограничения Индексы Зависимости Триггеры Данные Master/Detail View Описание Скрипт Права Протокол Сравнение Задачи

KOD_TR INTEGER NOT NULL

#	PK	BK	UNQ	Название	Тип	Дополн	Длина	Точ...	Подтип	Масштаб	Не ну...	Кодировка	Коллате	Описан...	Вычислить как...	По умолчанию
1				KOD_SPSEC	INTEGER						☒					
2				NAME_SPSEC	VARCHAR		100				☐	NONE	NONE			
3				DNI	VARCHAR		50				☐	NONE	NONE			
4				STOIM	NUMERIC		15	2			☐					
5				KOD_TR	INTEGER						☒					

Описание поля Объекты, использующие поле

Рисунок 3.5 – Довідник спортивних секцій (spsec)

Таблица : [ABON] : D:\Gym\db\GYM.GDB (D:\Gym\db\GYM.GDB)

Количество записей: 1 Type: Persistor ABON

Поля: Ограничения Индексы Зависимости Триггеры Данные Master/Detail View Описание Скрипт Права Протокол Сравнение Задачи

KOD_ABON INTEGER NOT NULL

#	PK	BK	UNQ	Название	Тип	Дополн	Длина	Точ...	Подтип	Масштаб	Не ну...	Кодировка	Коллате	Описан...	Вычислить как...	По умолчанию
1				KOD_ABON	INTEGER						☒					
2				KOD_KL	INTEGER						☒					
3				KOD_SPSEC	INTEGER						☒					
4				DATA_POK	DATE						☐					
5				VID_1	VARCHAR		1				☐	NONE	NONE			
6				VID_2	VARCHAR		1				☐	NONE	NONE			
7				VID_3	VARCHAR		1				☐	NONE	NONE			
8				VID_4	VARCHAR		1				☐	NONE	NONE			
9				VID_5	VARCHAR		1				☐	NONE	NONE			
10				VID_6	VARCHAR		1				☐	NONE	NONE			
11				VID_7	VARCHAR		1				☐	NONE	NONE			
12				VID_8	VARCHAR		1				☐	NONE	NONE			
13				VID_9	VARCHAR		1				☐	NONE	NONE			
14				VID_10	VARCHAR		1				☐	NONE	NONE			
15				VID_11	VARCHAR		1				☐	NONE	NONE			
16				VID_12	VARCHAR		1				☐	NONE	NONE			

Описание поля Объекты, использующие поле

Рисунок 3.6 – Абонементи (abon)

3.4.2 Створення ключів

Створення первинних та зовнішніх ключів таблиць бази даних Gym.gdb показано на рисунках 3.7-3.12.

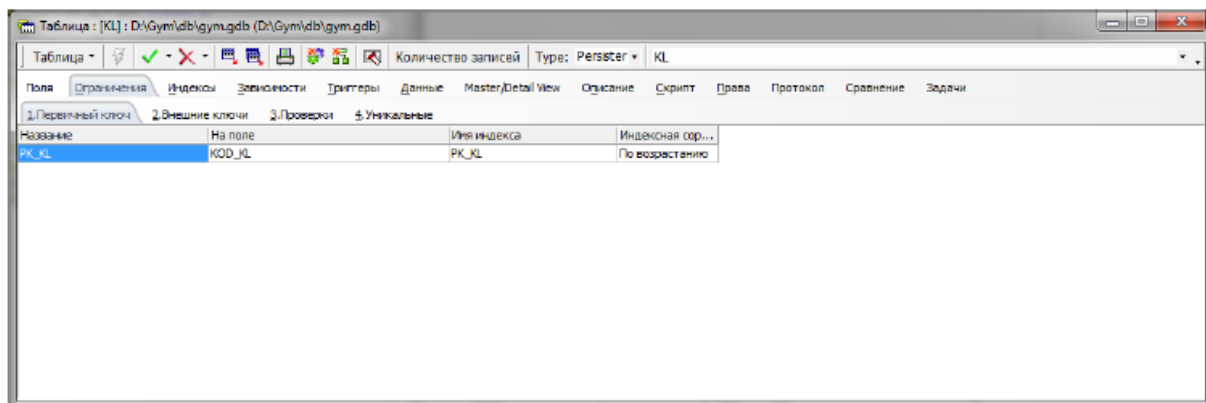


Рисунок 3.7 – Створення первинного ключа для Довідник клієнтів (kl)

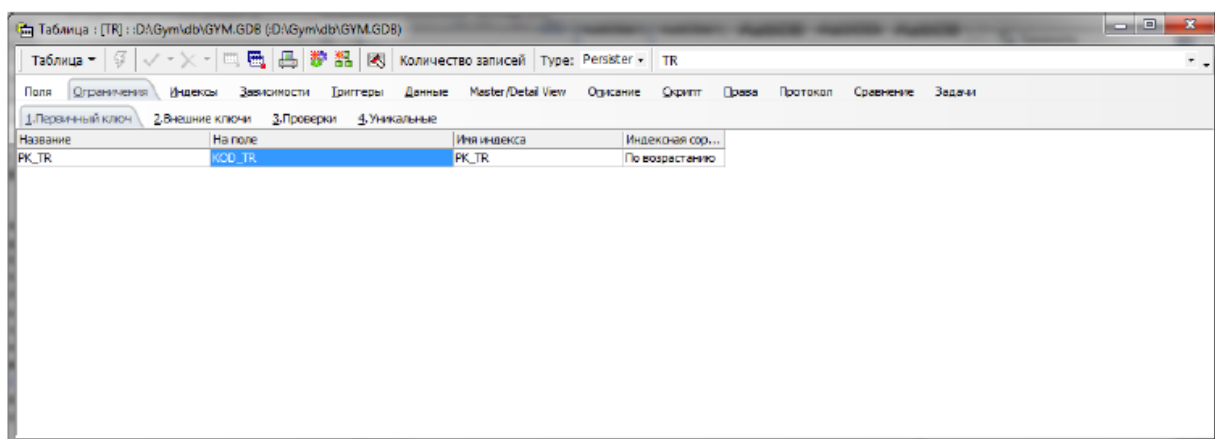


Рисунок 3.8 – Створення первинного ключа для Довідник тренерів (tr)

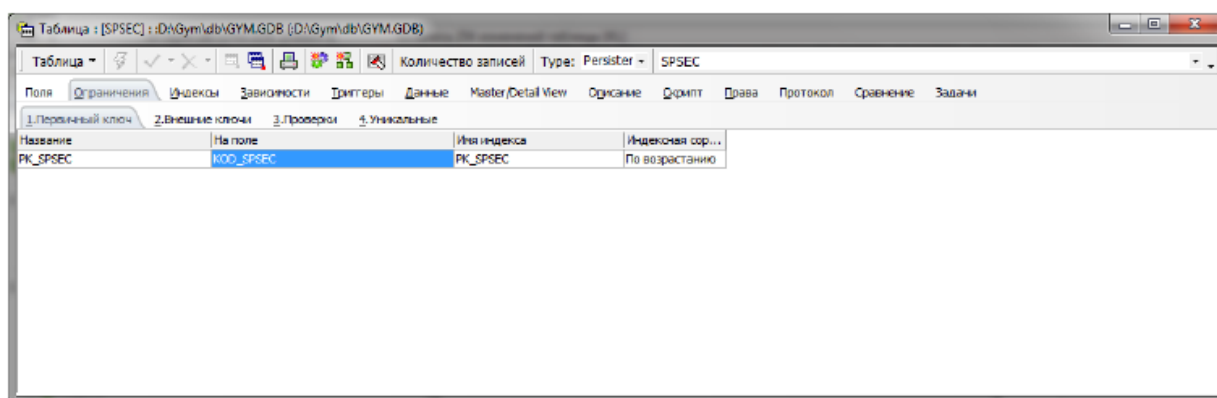


Рисунок 3.9 – Створення первинного ключа для Довідник спортивних секцій (spsec)

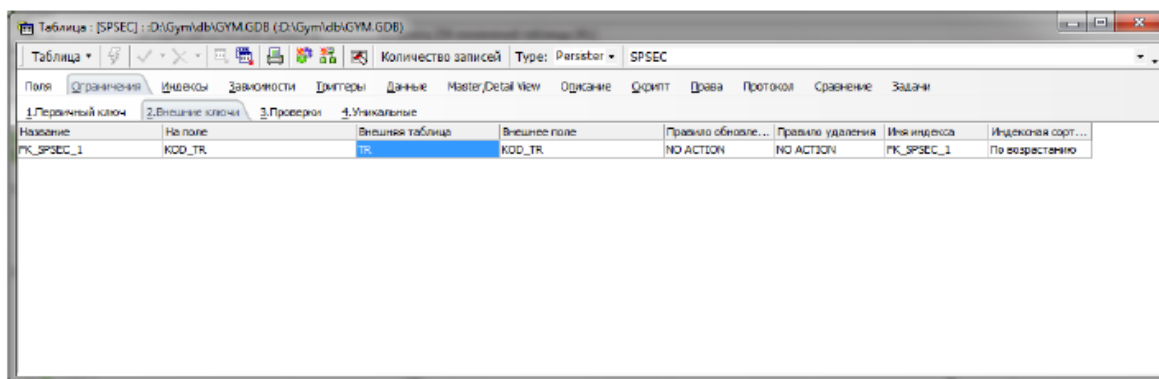


Рисунок 3.10 – Створення зовнішнього ключа для Довідник спортивних секцій (spsec)

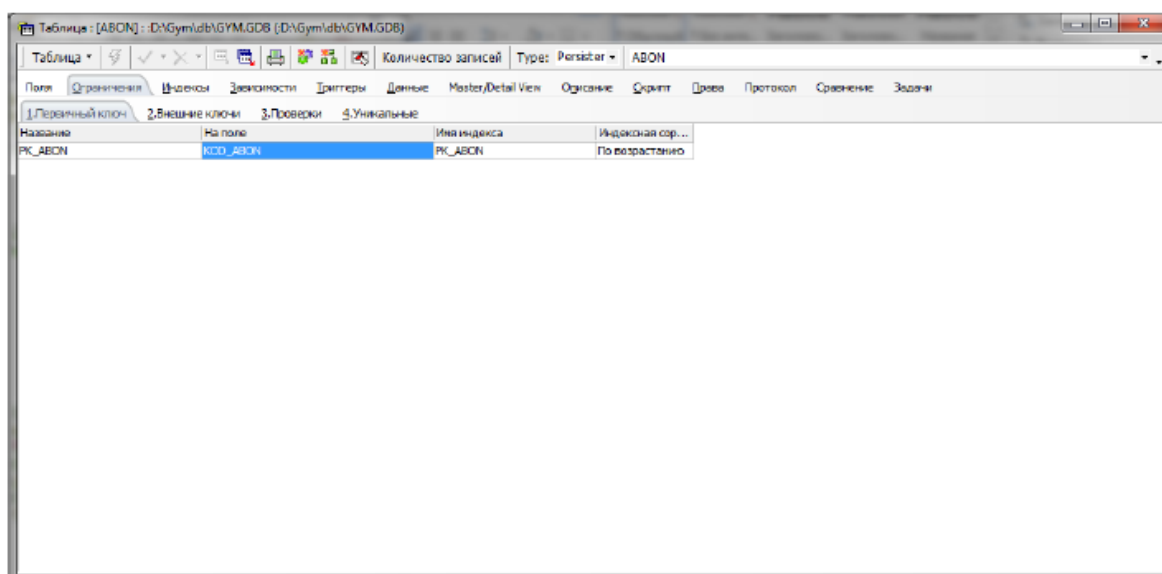


Рисунок 3.11 – Створення первинного ключа для Абонементи (abon)

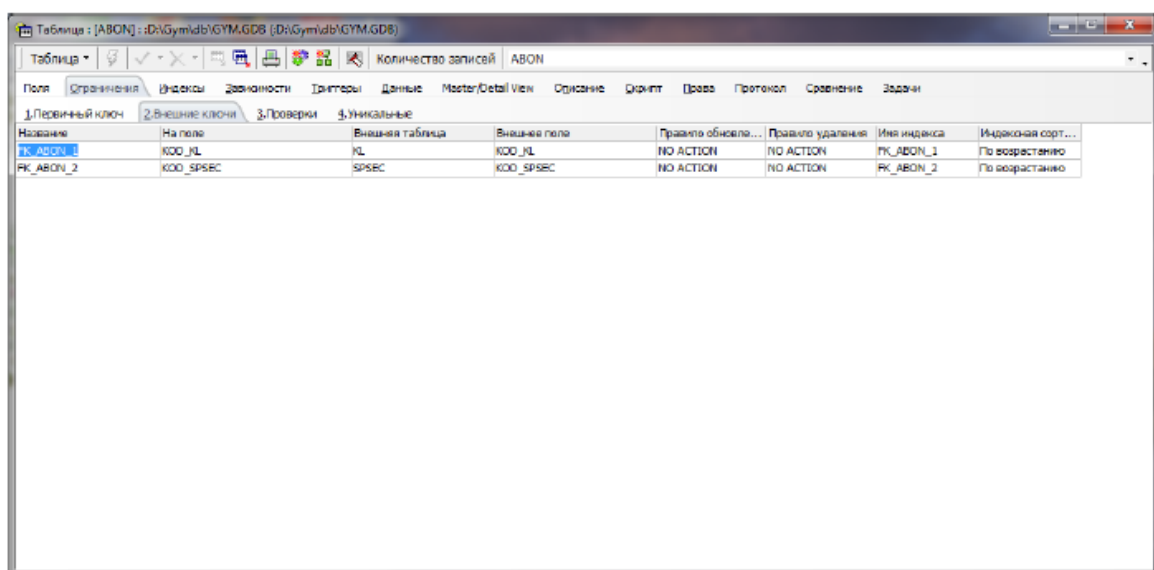


Рисунок 3.12 – Створення зовнішніх ключів для Абонементи (abon)

3.4.3 Створення процедур

Створення процедур на додавання, видалення та редагування даних в таблицях бази даних Gym.gdb показано на рисунках 3.13-3.24.

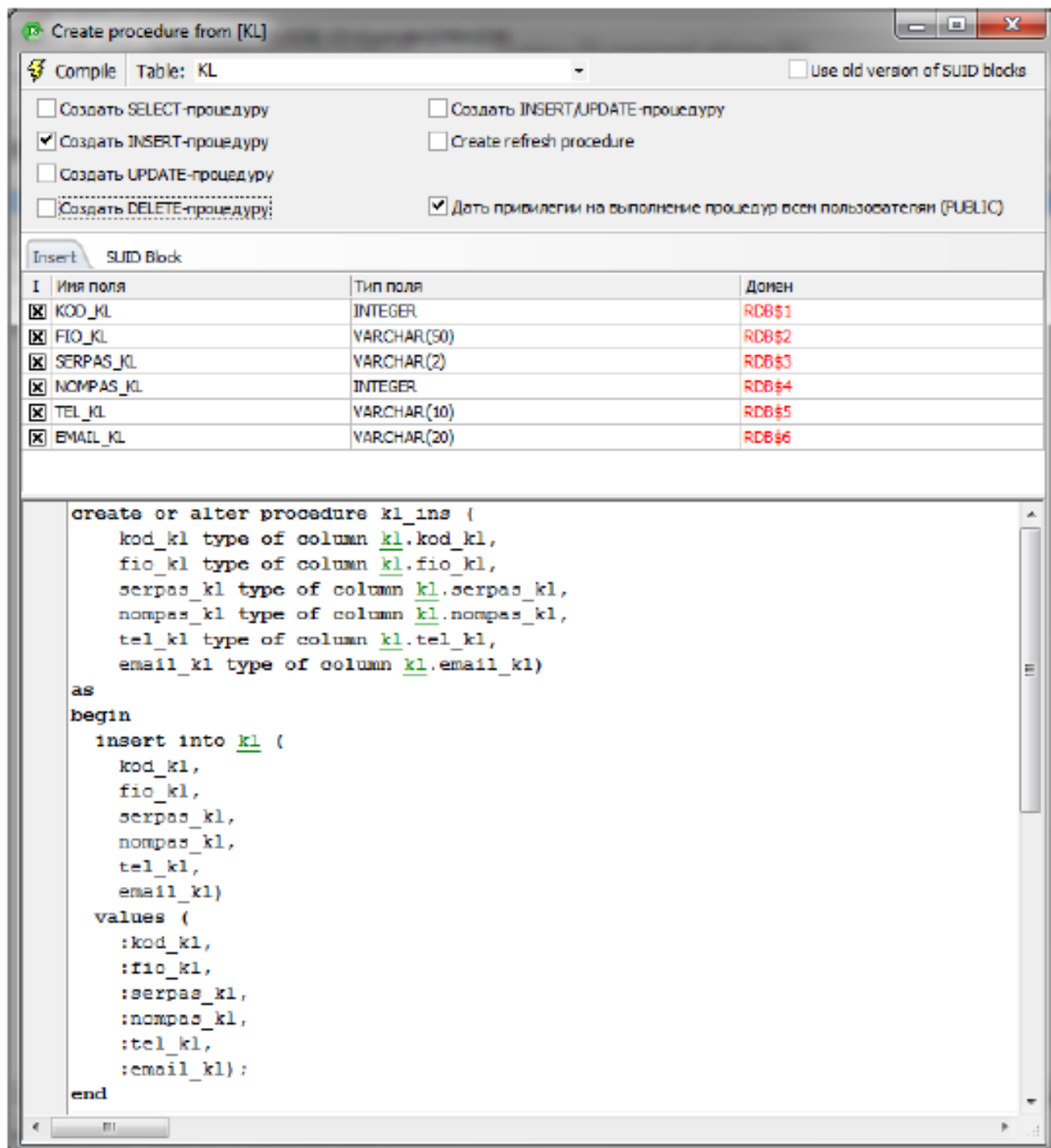


Рисунок 3.13 – Процедура на додавання для Довідник клієнтів (kl)

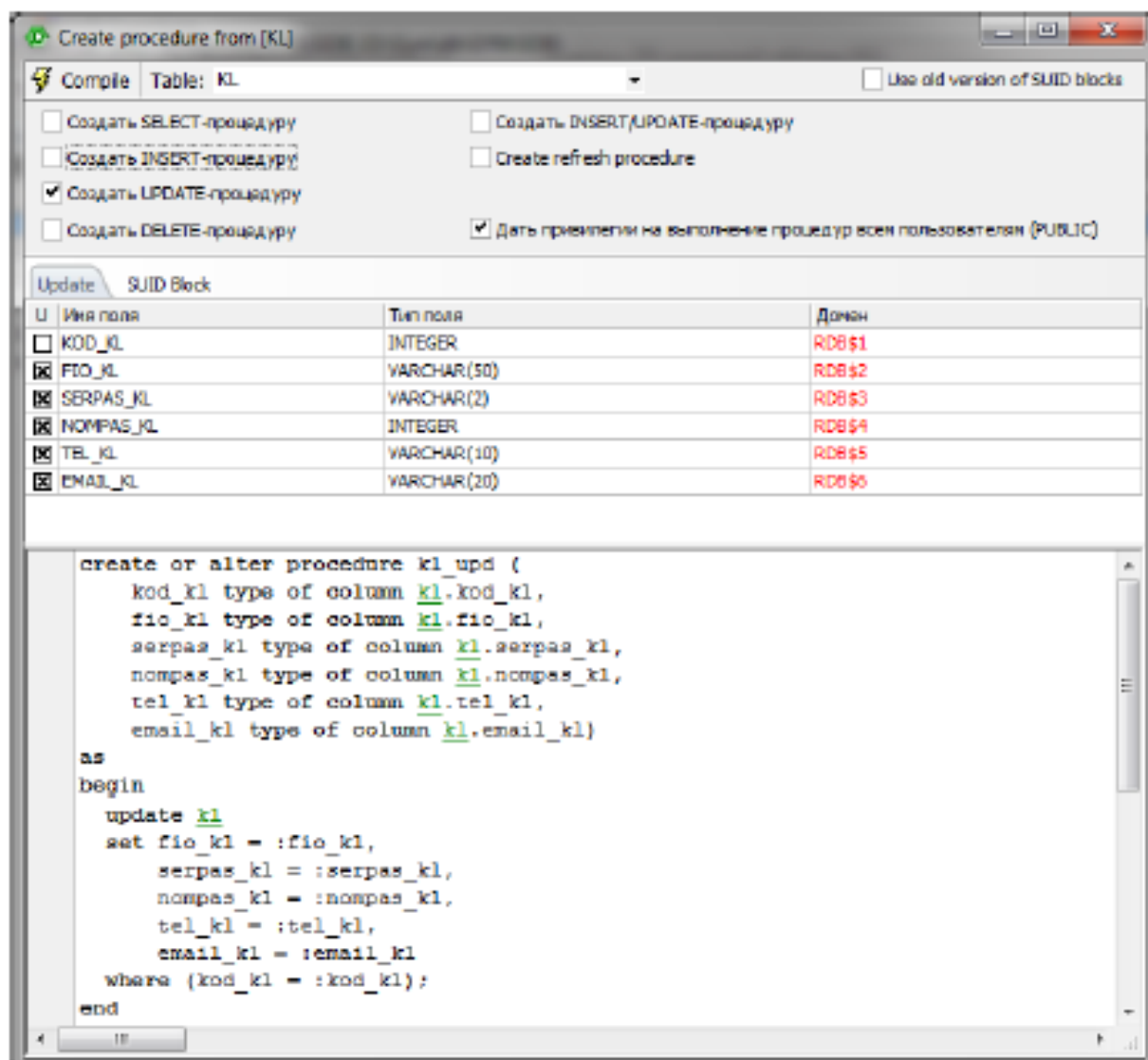


Рисунок 3.14 – Процедура на редагування для Довідник клієнтів (kl)

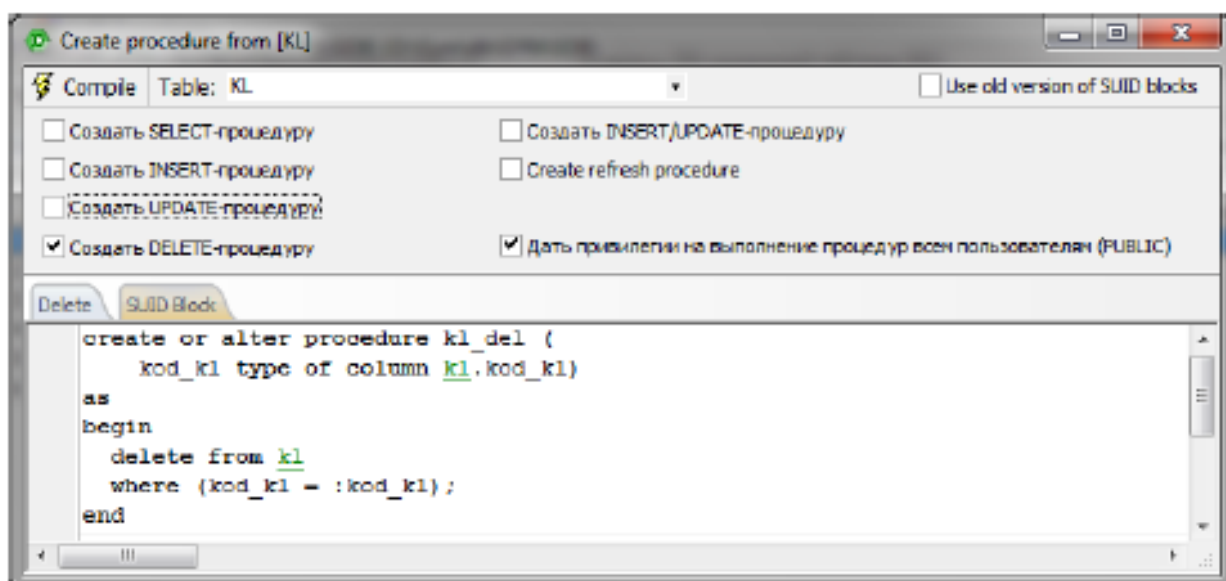


Рисунок 3.15 – Процедура на видалення для Довідник клієнтів (kl)

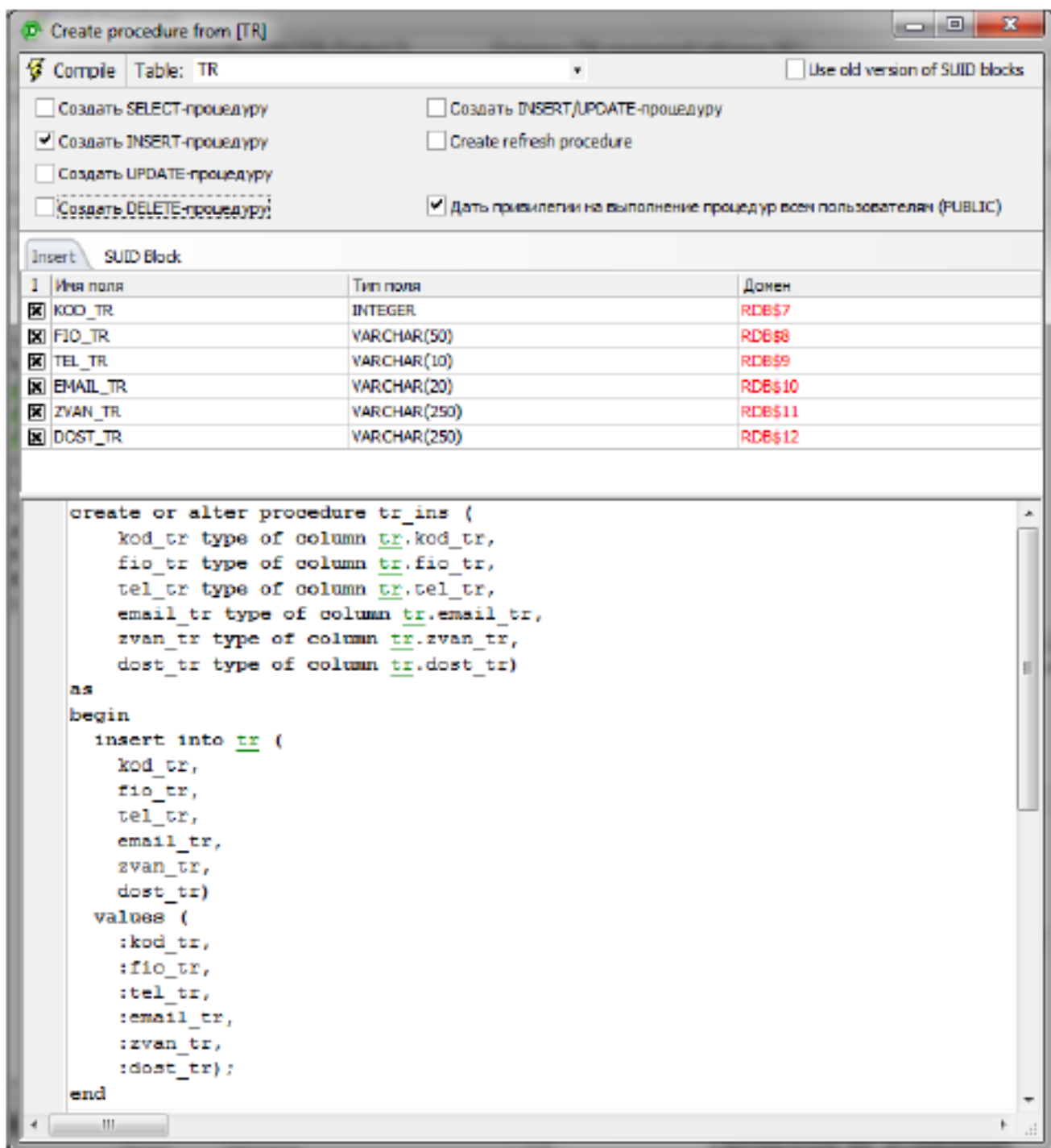


Рисунок 3.16 – Процедура на додавання для Довідник тренерів (tr)

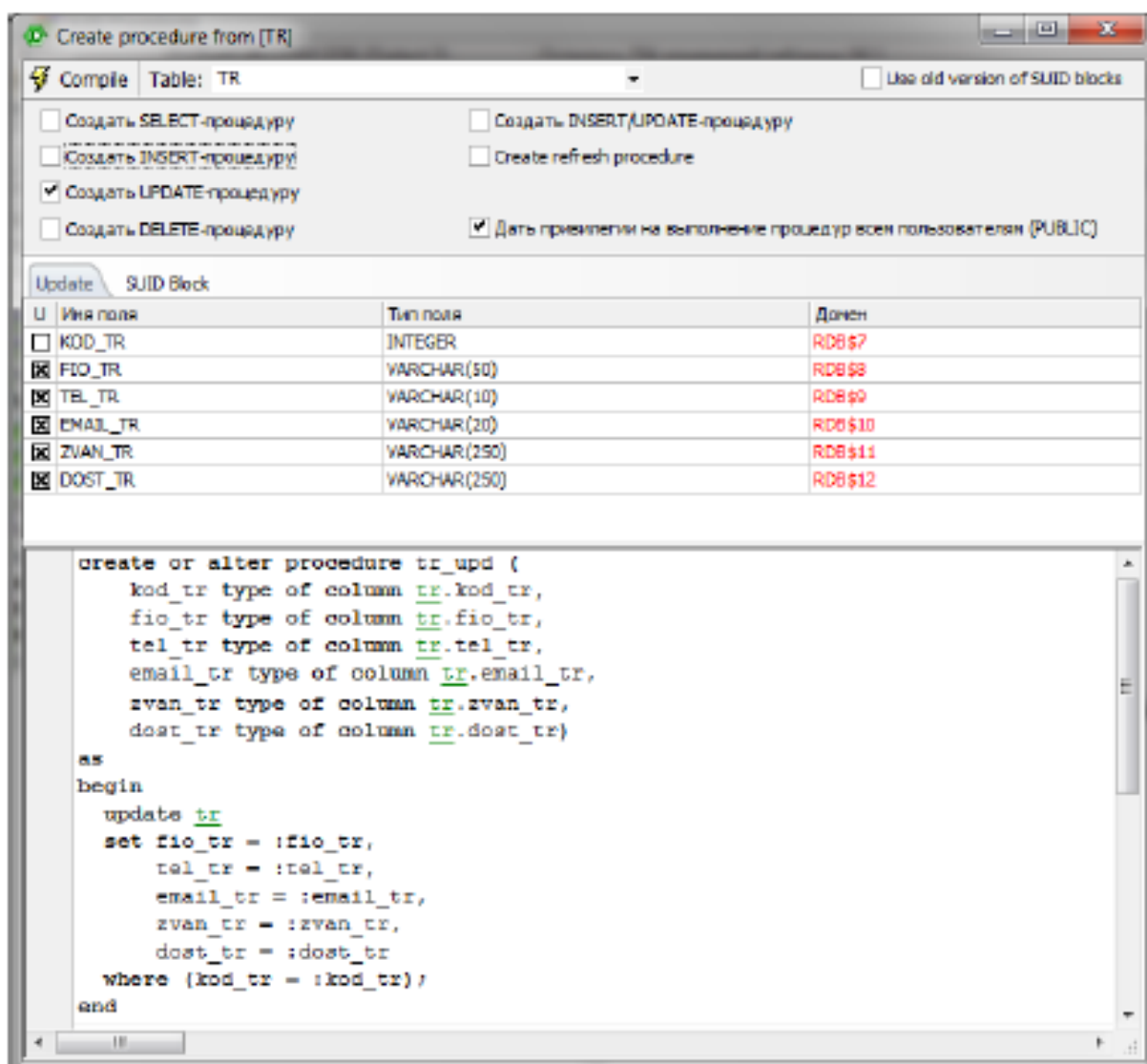


Рисунок 3.17 – Процедура на редагування для Довідник тренерів (tr)

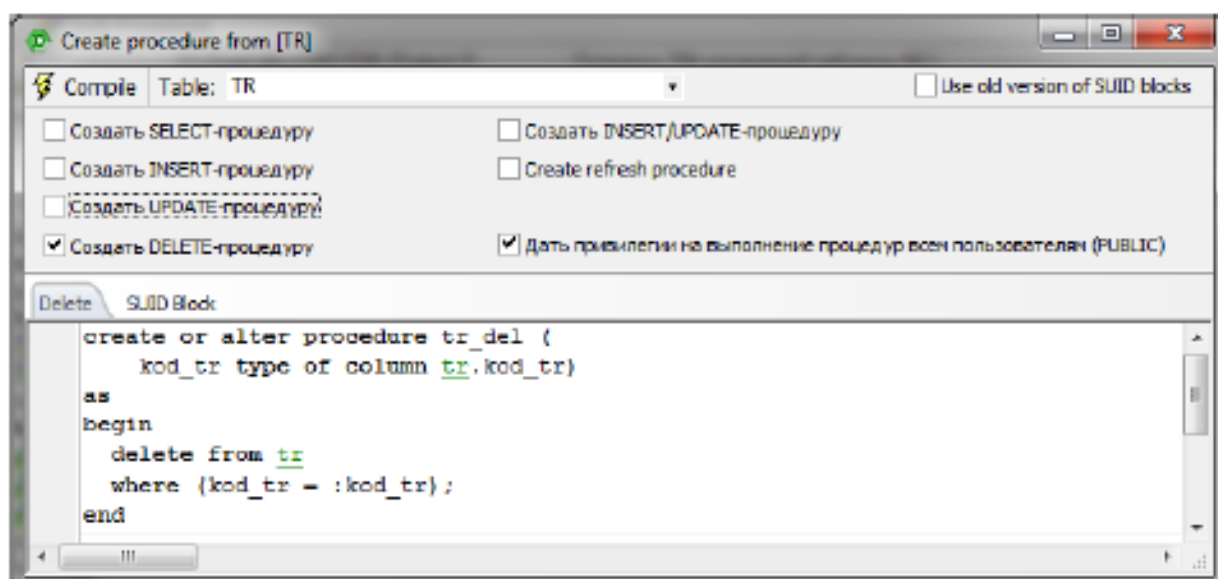


Рисунок 3.18 – Процедура на видалення для Довідник тренерів (tr)

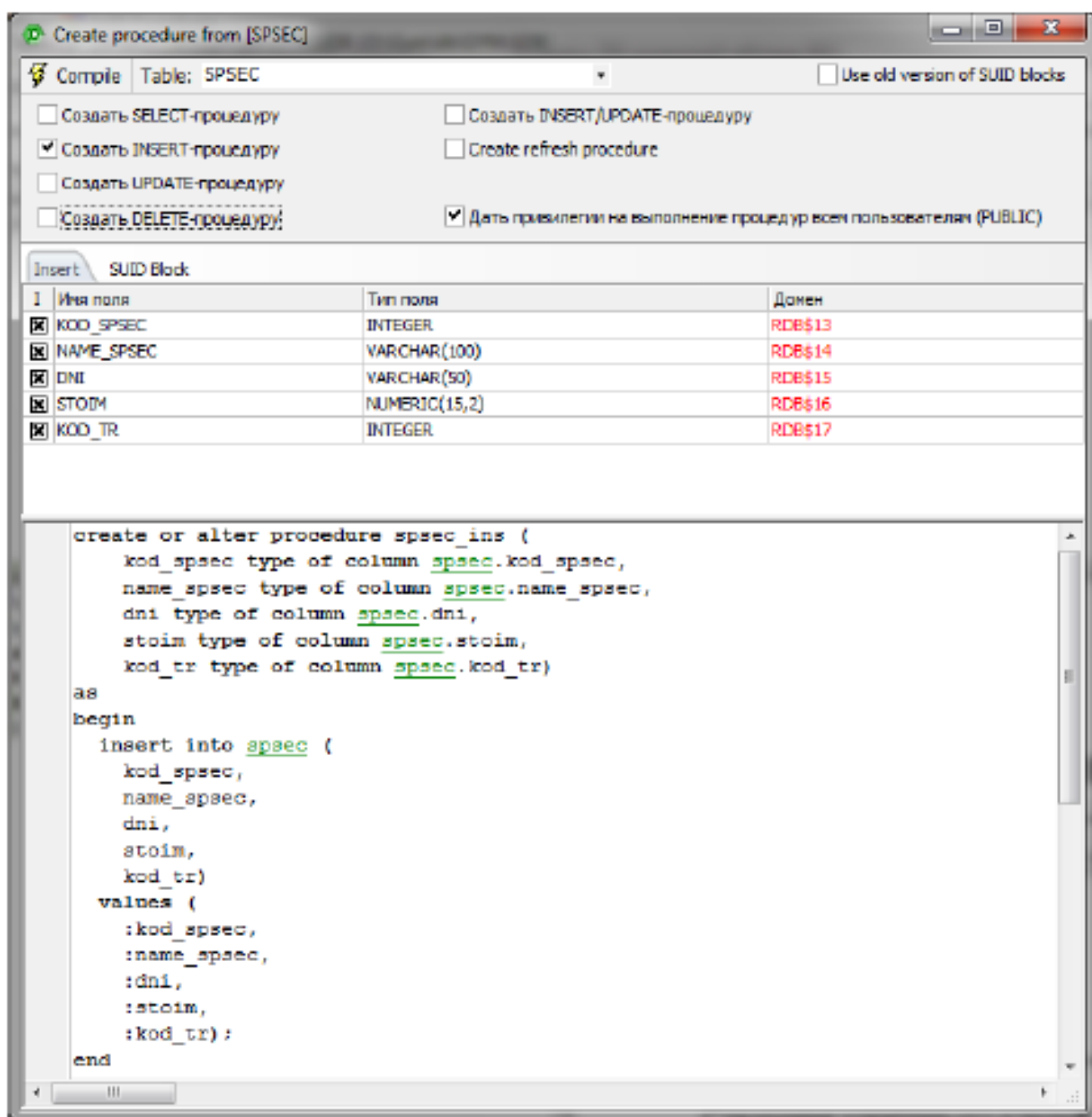


Рисунок 3.19 – Процедура на додавання для Довідник спортивних секцій (spsec)

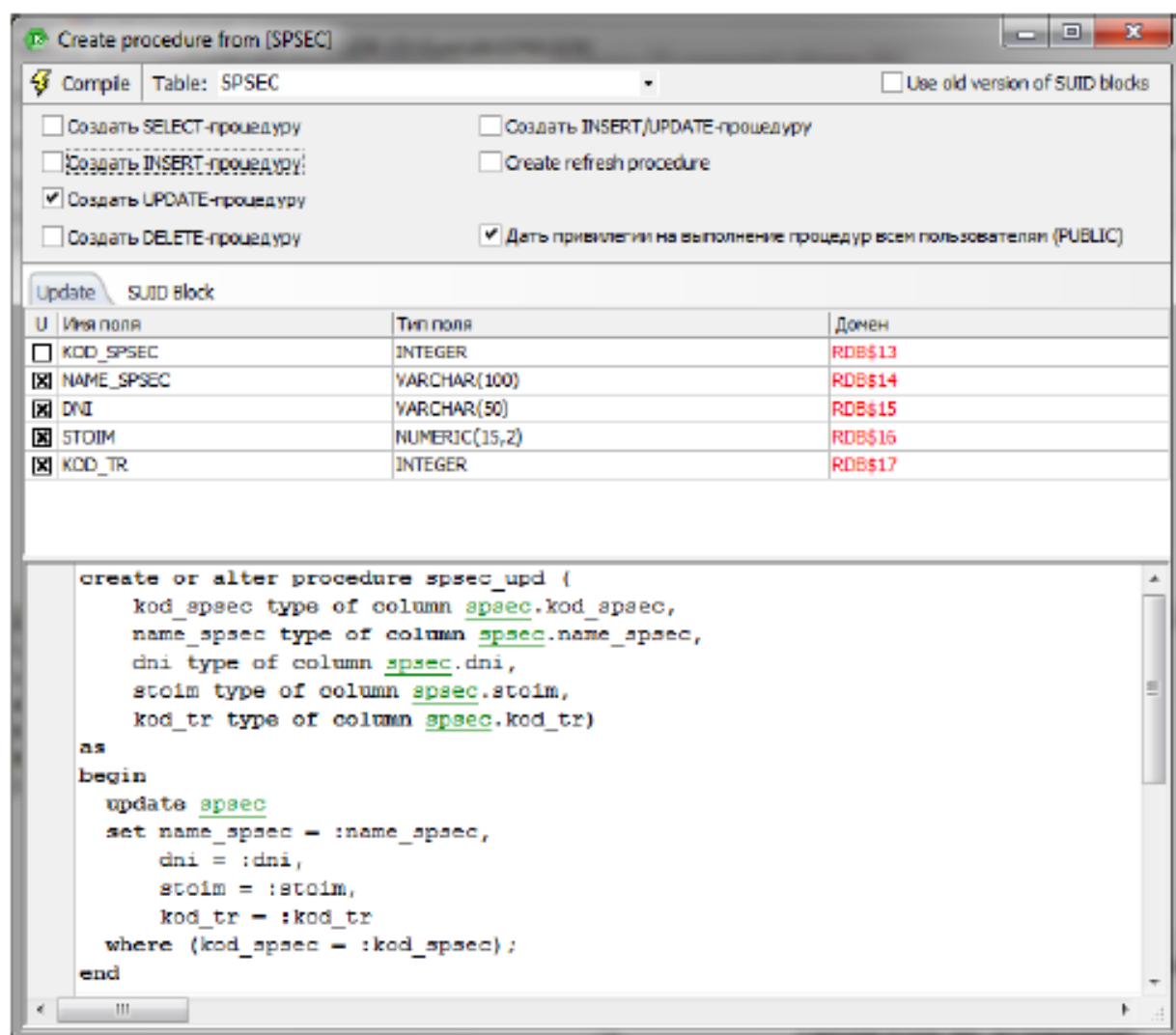


Рисунок 3.20 – Процедура на редагування для Довідник спортивних секцій (spsec)

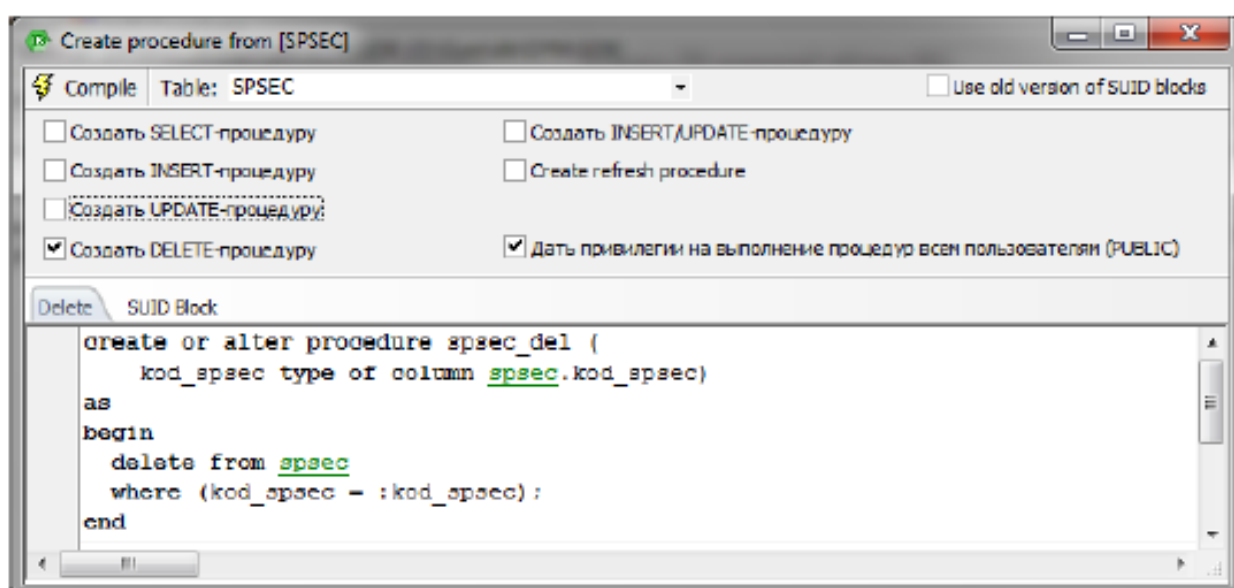


Рисунок 3.21 – Процедура на видалення для Довідник спортивних секцій (spsec)

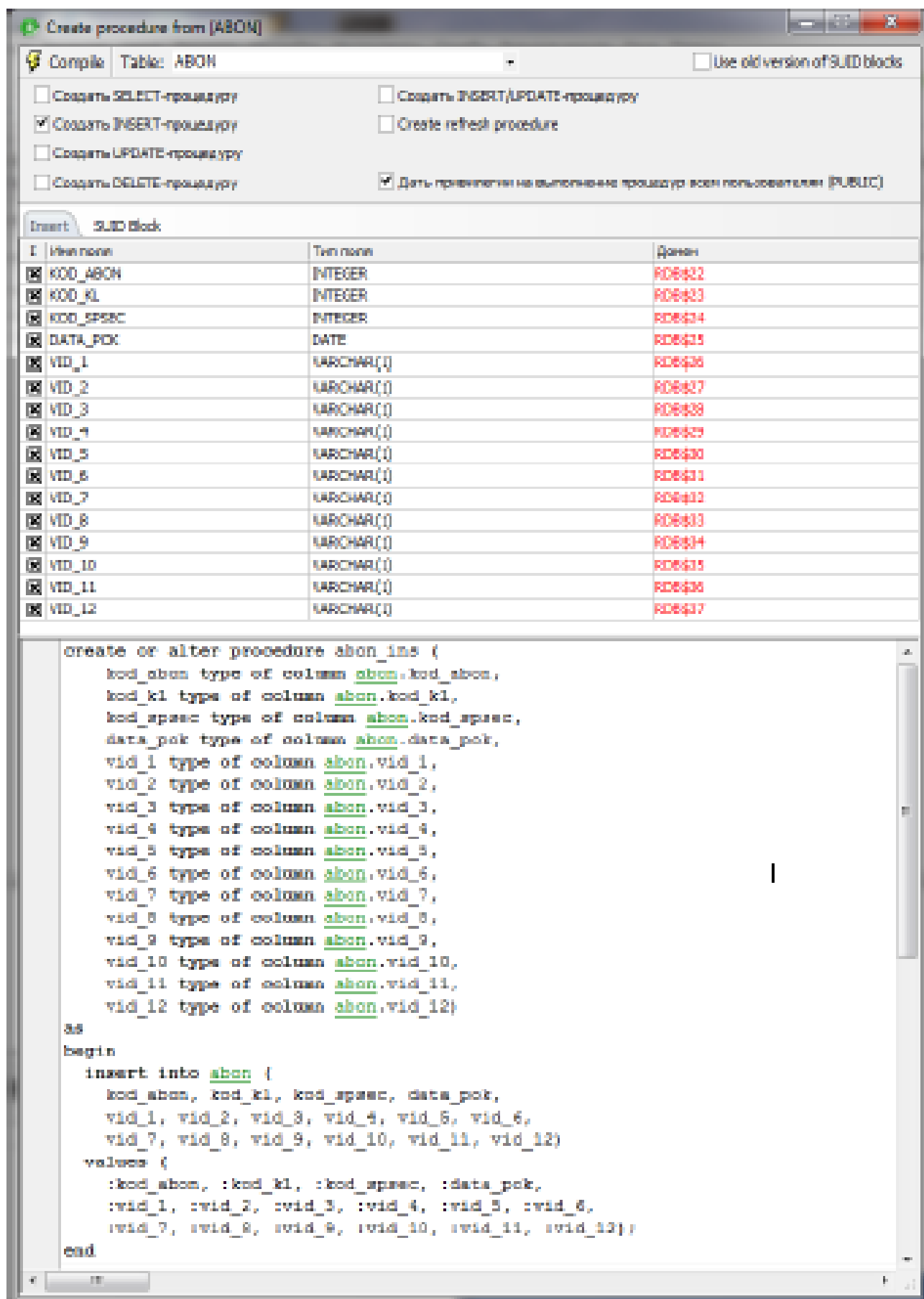


Рисунок 3.22 – Процедура на додавання для Абонементи (abon)

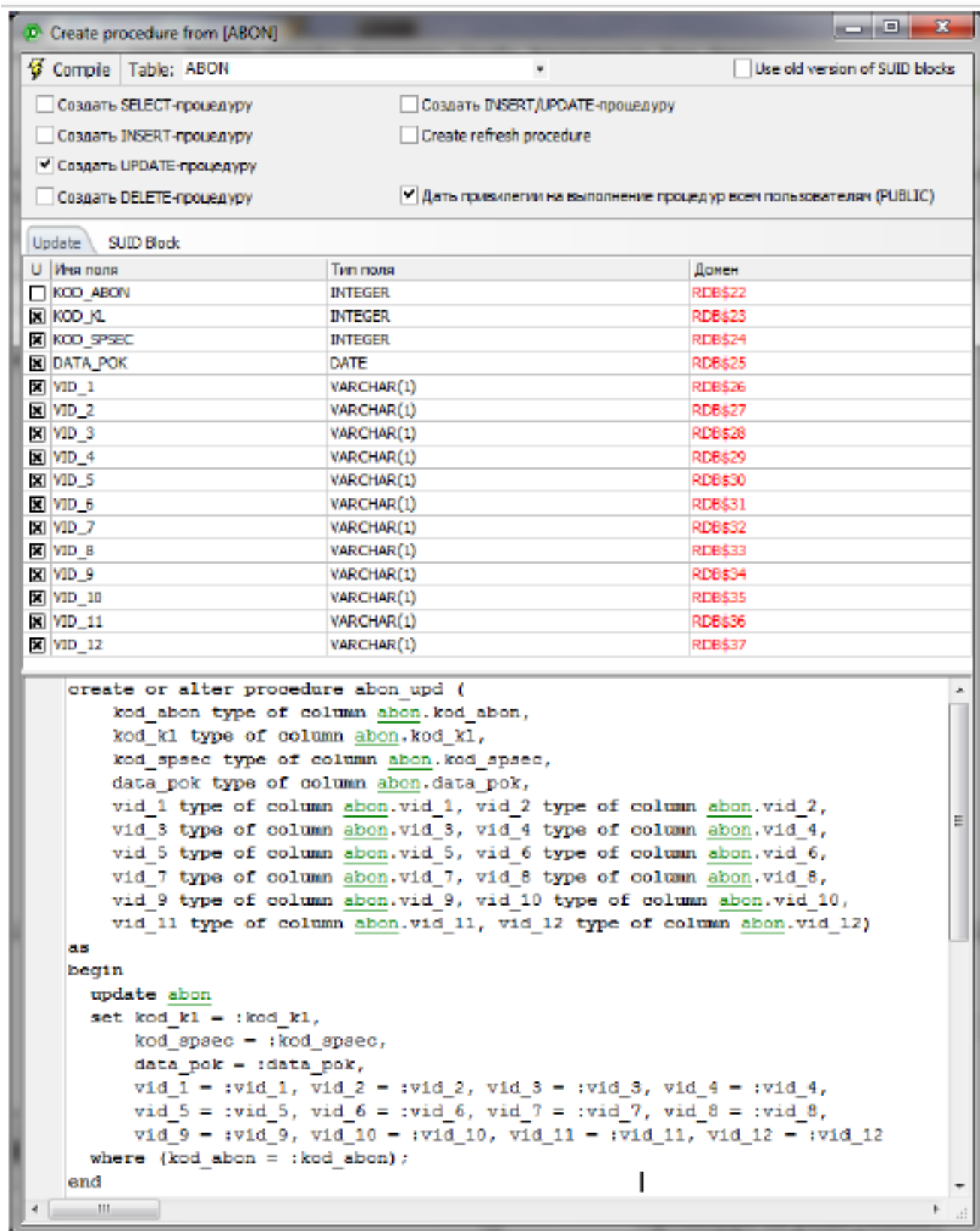


Рисунок 3.23 – Процедура на редагування для Абонементи (abon)

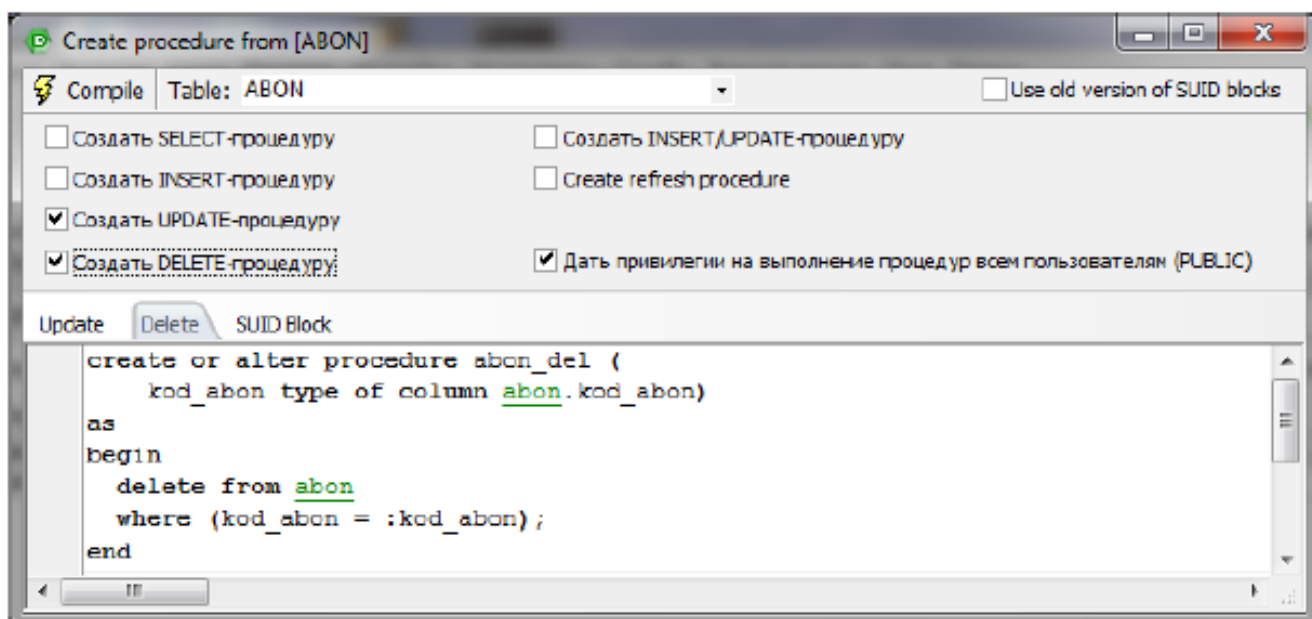


Рисунок 3.24 – Процедура на видалення для Абонементи (abon)

3.4.4 Створення діаграми бази даних

На рисунку 3.25 зображено структуру бази даних Gym.gdb. Основною перевагою реляційних баз даних є можливість зберігання тематично поділеної інформації у кількох таблицях із подальшим встановленням між ними логічних зв'язків, що дозволяє об'єднати їх в єдину цілісну систему. Однак такі зв'язки не створюються автоматично – їх необхідно формувати вручну.

Зв'язки між таблицями встановлюються через відповідні поля і формуються попарно. При цьому одна з таблиць виступає як головна, а інша – як підлегла. Цікаво, що одна й та сама таблиця може мати різний статус у різних зв'язках: бути головною в одному і підлеглою в іншому.

Зазвичай поле, за яким здійснюється зв'язок у головній таблиці, є первинним ключем. У підлеглий таблиці для встановлення зв'язку може використовуватися звичайне поле. При цьому важливо не стільки, щоб назви полів збігалися, скільки те, щоб вони мали однаковий тип даних та відповідні розміри.

У реляційних базах даних застосовуються два основні типи зв'язків між таблицями:

- "Один до одного" (1:1) – кожен запис головної таблиці відповідає одному запису у підлеглий. У такому разі обидві таблиці з'єднуються за первинними ключами.
- "Один до багатьох" (1:M) – один запис з головної таблиці може відповідати кільком записам у підлеглий таблиці. При цьому первинний ключ головної таблиці зв'язується з непервинним полем у підлеглий.

Такий підхід дозволяє будувати гнучкі, масштабовані інформаційні структури, що ідеально підходять для систем обліку та управління, як-от спортивні комплекси чи оздоровчі заклади.

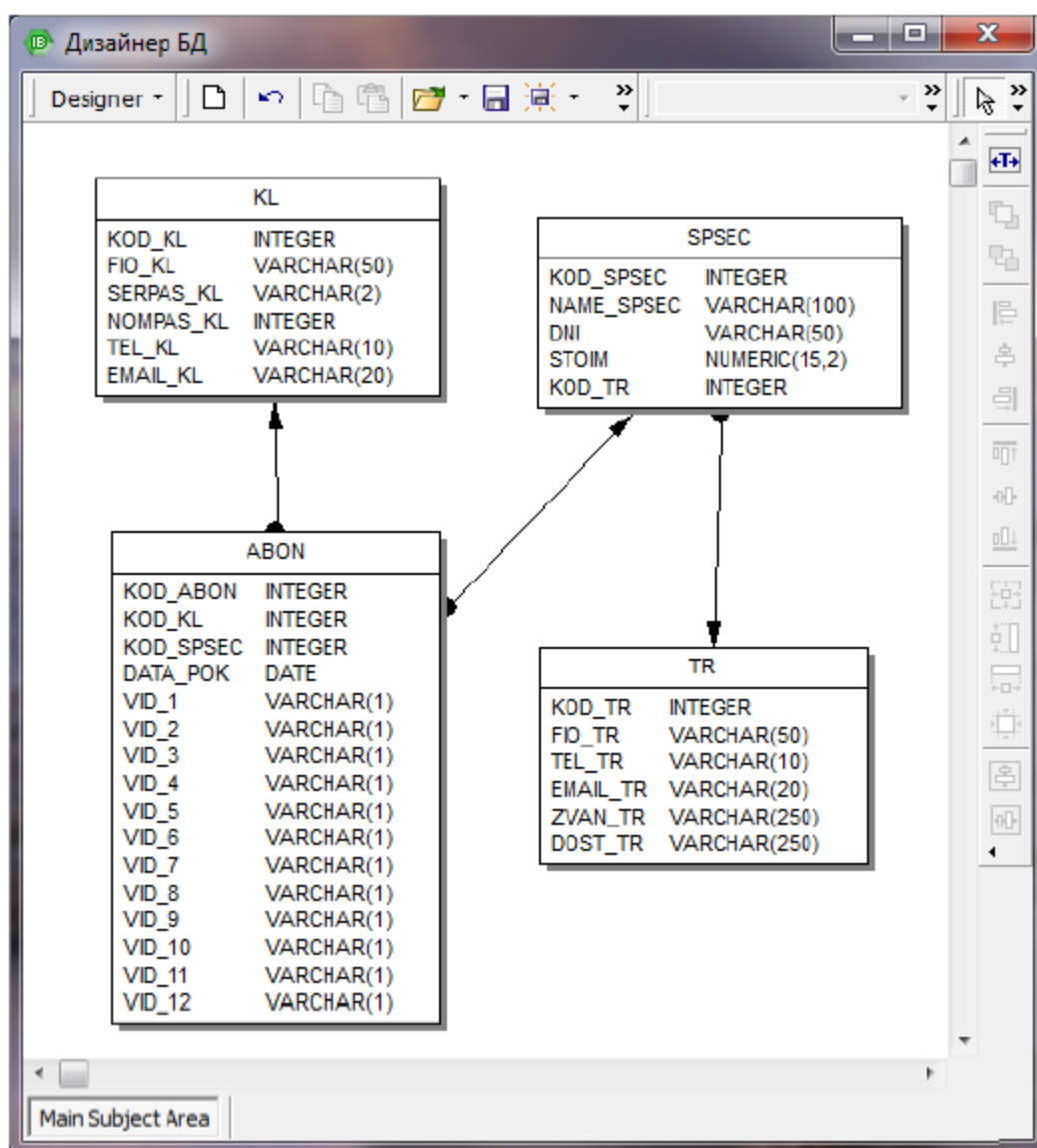


Рисунок 3.25 – Діаграма бази даних Gym.gdb

Якщо поле, яке використовується для встановлення зв'язку між таблицями, є унікальним ключем як у головній, так і в підлеглій таблиці, SQL Server формує між ними зв'язок типу "один до одного" (1:1). У такому разі кожному запису в головній таблиці відповідає лише один запис у підпорядкованій.

У випадку, коли ключове поле міститься тільки в головній таблиці, а в підпорядкованій – не є ключовим або входить до складу складного ключа, система інтерпретує цей зв'язок як "один до багатьох" (1:M). Тобто, одному запису в головній таблиці може відповідати кілька записів у підпорядкованій.

Роль головної та підлеглої таблиць. Важливо правильно розуміти, чим відрізняються головна та підпорядкована таблиці. Основна різниця полягає не у структурі, а у способі, яким Система управління базами даних (СУБД) забезпечує цілісність даних у разі змін або видалень записів.

Принципи забезпечення цілісності:

- У підпорядковану таблицю не можна додати запис, якщо відповідне значення ключа відсутнє в головній таблиці.
- Неможливо вилучити запис із головної таблиці, якщо на нього посилаються записи з підпорядкованої.
- Якщо змінюється значення ключа у головній таблиці, ці зміни автоматично мають бути синхронізовані з усіма відповідними записами в підлеглій таблиці.

Ці механізми гарантують логічну узгодженість даних, запобігаючи появі «висячих» посилань і структурних помилок у реляційній базі даних.

3.5 Екранні форми інформаційної системи

3.5.1 Нормативно-довідникова інформація

Екранні форми, що містять нормативно-довідникову інформацію показані на рисунках 3.26-3.31.

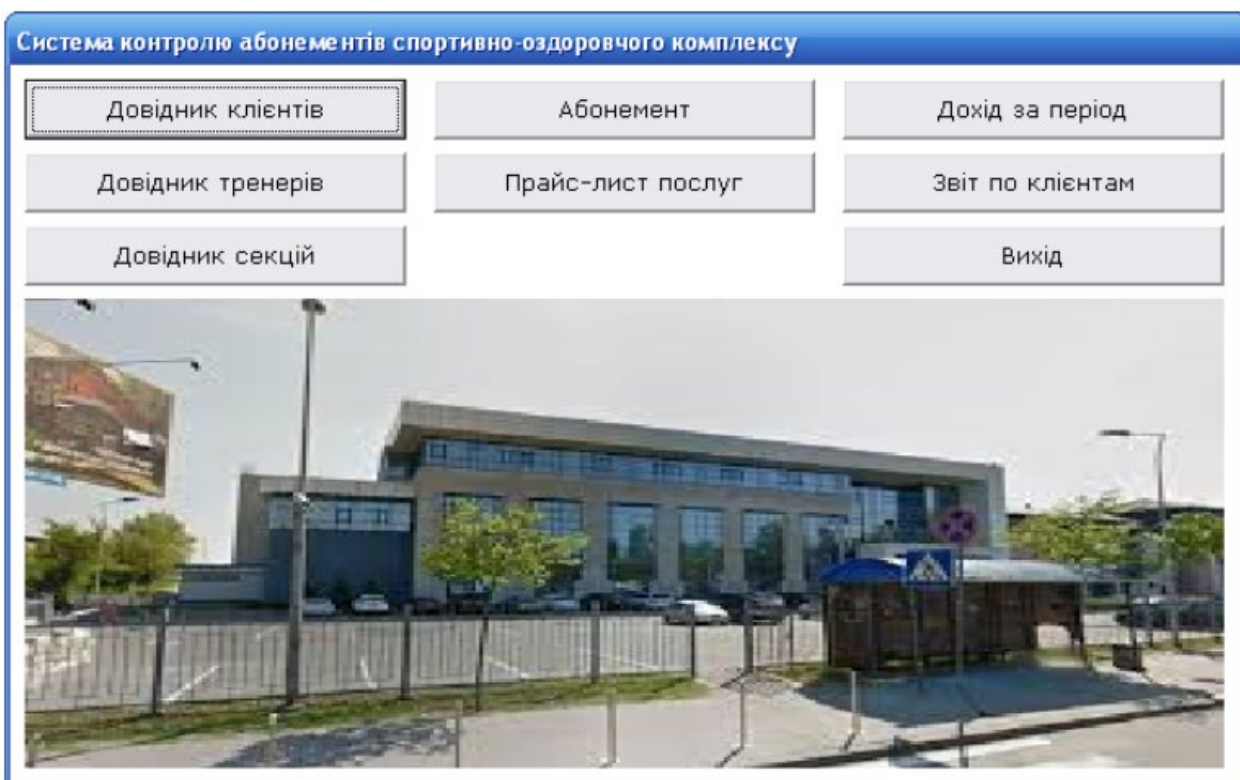


Рисунок 3.26 – Головна форма програми

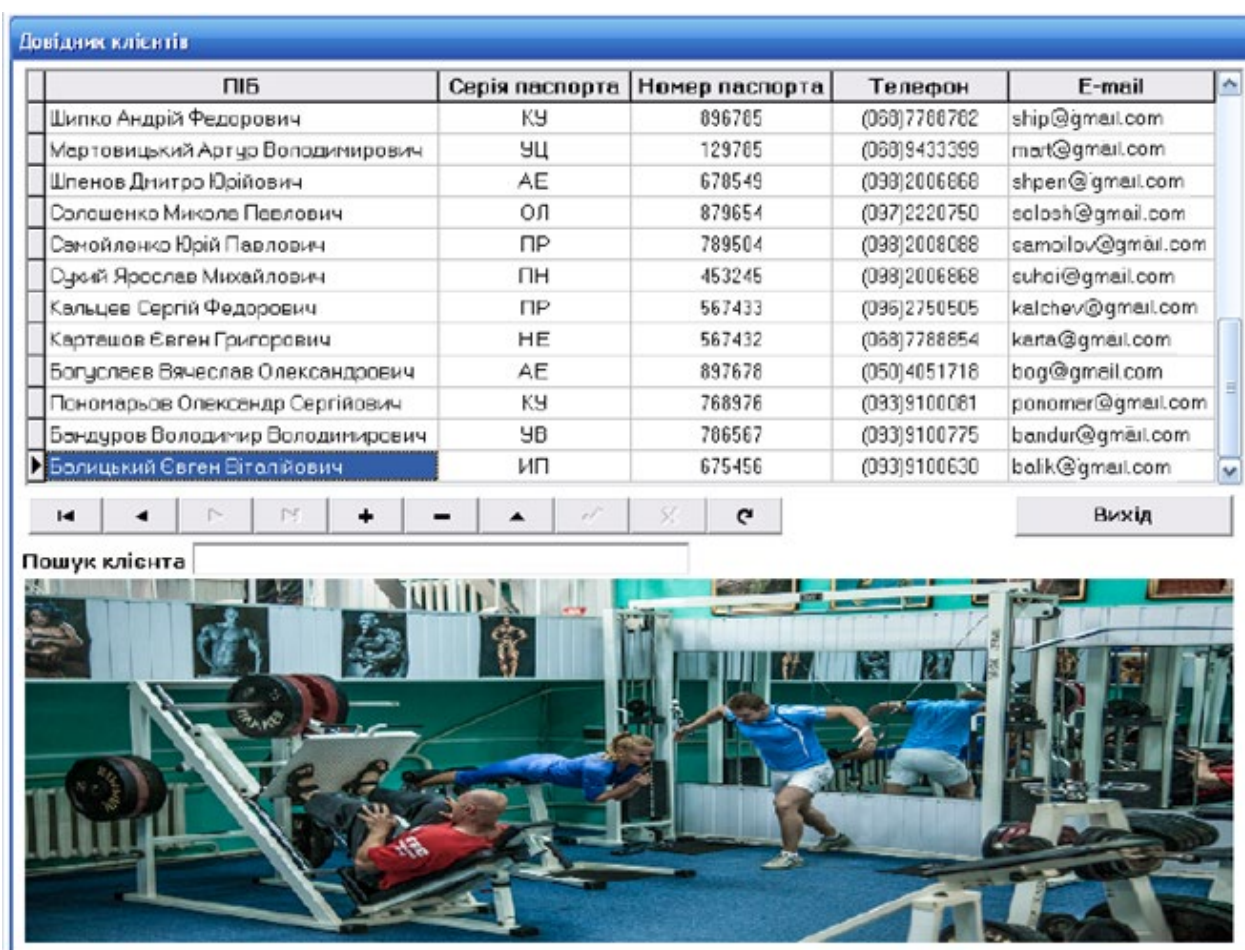


Рисунок 3.27 – Довідник клієнтів

Довідник клієнтів

ПІБ	Серія паспорта	Номер паспорта	Телефон	E-mail
Бутківський Віктор Володимирович	АП	342342	(096)6699951	butik@gmail.com
Любоненко Юрій Вікторович	АЕ	820619	(096)3531113	lubon@gmail.com
Богуслав В'ячеслав Олександрович	АЕ	897678	(050)4051718	bog@gmail.com
Бандуров Володимир Володимирович	УВ	786567	(093)9100775	bandur@gmail.com
Балицький Євген Віталійович	ІП	675456	(093)9100630	balik@gmail.com

Пошук клієнта



Рисунок 3.28 – Пошук клієнта в Довіднику клієнта

Довідник тренерів

Тренери		
ПІБ	Телефон	E-mail
Костенко Микола Петрович	096 0666099	kostenko.mykola@gmail.com
Прохніч Валерій Миколайович	097 7770900	
Корзей Михайло Євгенович	097 0997979	
Федоріна Наталія Борисівна	098 0999008	teslja_n@gmail.com
Бурко Сергій Валерійович	099 1011191	ser_bur01@gmail.com
Муратов Вахабжан Аркелжанович	093 4633333	
Гордеева Світлана Валентинівна	096 2752444	

Звання
 майстер спорту України міжнародного класу з футболу

Досягнення
 чемпіон світу серед студентів з міні-футболу в складі збірної команди України



Рисунок 3.29 – Довідник тренерів

3.5.2 Вхідна інформація

Екранні форми, що містять вхідну інформацію представлені на рисунках 3.32-3.42.

Абонементи																	
Оформити новий абонемент		Пошук клієнта:		Повернути роздрукувати абонемент		Вид											
№	ПІБ клієнта	Назва секції	Вартість заняття на місяць, грн.	Дійсний до ...	Дні відвідувань	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
22	Васильев Александр Андреевич	рукопашний бій	600	11.04.2025	Пн, Ср, Пт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
23	Глазунов Сергей Миколайович	спортивна аеробіка	600	19.04.2025	Вт, Чт, Сб	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
24	Шипко Андрей Федорович	теніс настільний	600	01.05.2025	Пн, Ср, Пт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
25	Глазунов Сергей Миколайович	теніс настільний	600	01.05.2025	Пн, Ср, Пт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
26	Балацкий Евгений Витальевич	спортивне орієнтування	300	12.05.2025	Пн, Ср, Пт	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
27	Мартовичский Артур Володимирович	міні-футбол (чоловіки)	600	30.05.2025	Вт, Чт, Сб	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
← → ↶ ↷ + - ⬆ ⬇ ⬆																	

Рисунок 3.32 – Абонемент

Оформити новий абонемент

Обрати клієнта

Обрати секцію

Дні відвідувань

Вартість заняття на місяць, грн.

Зберегти

Відмінити

Рисунок 3.33 – Оформити новий абонемент

Довідник клієнтів

ПІБ	Серія паспорта	Номер паспорта	Телефон	E-mail
Шипко Андрій Федорович	КУ	896785	(068)7788782	ship@i.ua
Мертовийський Артур Володимирович	УЦ	129785	(068)9433399	mart@mail.ru
Шпенюк Дмитро Юрійович	АЕ	678549	(098)2006888	shpen@i.ua
Солошенко Микола Павлович	ОЛ	879654	(097)2220750	solosh@gmail.com
Самойленко Юрій Павлович	ПР	789504	(098)2008088	samoilov@mail.ru
Сукач Ярослав Михайлович	ПН	453245	(098)2006888	suha@gmail.com
Кальцев Сергій Федорович	ПР	567433	(096)2750505	kalchev@mail.ua
Картешов Євген Григорович	НЕ	567432	(068)7788854	kara@mail.ru
Богуславський Вячеслав Олександрович	АЕ	897678	(050)4051718	bog@gmail.com
Понімарьов Олександр Сергійович	КУ	768978	(093)9100061	ponomer@i.ua
Бандуров Володимир Володимирович	УВ	786567	(093)9100775	bandur@mail.ru
Балицький Євген Віталійович	ІП	675456	(093)9100630	balik@i.ua

Пошук клієнта



Обрати клієнта Вихід

Рисунок 3.34 – Обрати клієнта

Довідник секцій

Назва секції	Дні відвідувань	Вартість заняття на місяць, грн.
футбол (чоловіки)	Вт, Чт, Сб	900
міні-футбол (жінки)	Вт, Чт, Сб	800
міні-футбол (чоловіки)	Вт, Чт, Сб	800
айкідо	Пн, Ср, Пт	720
атлетична гімнастика (бодібілдинг)	Вт, Чт, Сб	570
рукопашний бій	Пн, Ср, Пт	630
самбо і бойове самбо	Вт, Чт, Сб	660
фітнес	Вт, Чт, Сб	750
Боротьба спортивна	Пн, Ср, Пт	690
танцювальна аеробіка	Вт, Чт, Сб	900

Тренер Обрати тренера Обрати секцію

Вихід



Рисунок 3.35 – Обрати секцію

Оформити новий абонемент

Обрати клієнта: Бандуров Володимир Володимирович

Обрати секцію: боротьба спортивна

Дні відвідувань: 690

Вартість занять на місяць, грн.: Пн, Ср, Пт

Зберегти Відмінити



Рисунок 3.36 – Зберегти новий абонемент

Абонемент

Абонемент № 31

ПІБ клієнта: Бандуров Володимир Володимирович

Назва секції: боротьба спортивна

Вартість занять на місяць, грн.: 690

Дійсний до ... 09.07.2025

Відвідування

☒ 1	☒ 2	☒ 3	☒ 4	☒ 5	☒ 6
☒ 7	☒ 8	☒ 9	☒ 10	☒ 11	☒ 12

Друк Вихід



Рисунок 3.37 – Новий абонемент

Абонементи

Оформити новий абонемент

Пошук клієнта:

Повторно роздрукувати абонемент

Вихід

№	ПІБ клієнта	Назва секції	Вартість заняття на місяць, грн.	Дійсний до ...	Дні відвідувань	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
25	Глазунов Сергій Миколайович	теніс настільний	600	01.05.2025	Пн, Ср, Пт		☒		☒								
26	Балицький Євген Віталійович	спортивне орієнтування	300	12.05.2025	Пн, Ср, Пт	☒	☒	☒									
27	Мартович Артур Володимирович	міні-футбол (чоловіки)	600	30.05.2025	Вт, Чт, Сб	☒											
28	Павлов Костянтин Юрійович	боротьба спортивна	690	25.05.2025	Пн, Ср, Пт												
30	Мартович Артур Володимирович	гирювий спорт	600	09.07.2025	Вт, Чт, Сб												
31	Бандуров Володимир Володимирович	боротьба спортивна	690	09.07.2025	Пн, Ср, Пт												

Рисунок 3.38 – Доданий абонемент

Абонементи

Оформити новий абонемент

Пошук клієнта:

Повторно роздрукувати абонемент

Вихід

№	ПІБ клієнта	Назва секції	Вартість заняття на місяць, грн.	Дійсний до ...	Дні відвідувань	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
26	Балицький Євген Віталійович	спортивне орієнтування	300	12.05.2025	Пн, Ср, Пт	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
31	Бандуров Володимир Володимирович	боротьба спортивна	690	09.07.2025	Пн, Ср, Пт	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Рисунок 3.39 – Пошук клієнта

Абонементи

Оформити новий абонемент

Пошук клієнта: бан

Повторно роздрукувати абонемент

Вихід

№	ПІБ клієнта	Назва секції	Вартість заняття на місяць, грн.	Дійсний до ...	Дні відвідувань	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
31	Бандуров Володимир Володимирович	боротьба спортивна	690	09.07.2025	Пн, Ср, Пт	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

<

Рисунок 3.40 – Відвідування клієнта

Абонементи

Абонемент №

ПІБ клієнта

Назва секції

Вартість занять на місяць, грн.

Дійсний до ...

Відвідування

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12



Рисунок 3.41 – Повторно роздрукувати абонемент при наявності змін

Preview - Tabular Information

Print Printer setup Scale Previous Next Stop Close

Прайс-лист послуг

Назва секції	Дні відвідувань	Вартість занять на місяць, грн.
баскетбол (жінки)	Пн, Ср, Пт	360
баскетбол (чоловіки)	Вт, Чт, Сб	450
боротьба вільна та греко-римська	Пн, Ср, Пт	600
важка атлетика	Пн, Ср, Пт	720
волейбол (жінки)	Вт, Чт, Сб	300
волейбол (чоловіки)	Пн, Ср, Пт	300
дзюдо	Вт, Чт, Сб	600
карате кіюдшин	Пн, Ср, Пт	750
легка атлетика	Вт, Чт, Сб	510
спортивне орієнтування	Пн, Ср, Пт	300
спортивна верховна	Вт, Чт, Сб	690
теніс настільний	Пн, Ср, Пт	600
фудбол (чоловіки)	Вт, Чт, Сб	900
мін-фудбол (жінки)	Вт, Чт, Сб	800
мін-фудбол (чоловіки)	Вт, Чт, Сб	800
аїкідо	Пн, Ср, Пт	720
атлетична гімнастика (бодібілдинг)	Вт, Чт, Сб	570
рукопашний бій	Пн, Ср, Пт	630
сендзю бойове самбо	Вт, Чт, Сб	660
фітнес	Вт, Чт, Сб	750
боротьба спортивна	Пн, Ср, Пт	690
танцювальна аеробіка	Вт, Чт, Сб	900

Page 1 of 1

Рисунок 3.42 – Прайс-лист послуг

3.5.3 Вихідна інформація

Екранні форма, що містить вихідну інформацію представлені на рисунках 3.43-3.47.



Рисунок 3.43 – Дохід за період

Звіт				
Дохід за період з 09.03.2025 по 09.06.2025				
№	ПІБ клієнта	Назва секції	Вартість зан	Дата придбання
24	Шипко Андрій Федорович	теніс настільний	600	01.04.2025
25	Глазунов Сергій Миколайович	теніс настільний	600	01.04.2025
26	Балицький Євген Віталійович	спортивне орієнтування	300	01.04.2025
27	Мартовницький Артур Володимирович	міні-футбол (чоловіки)	600	31.03.2025
28	Павлов Костянтин Юрійович	боротьба спортивна	600	25.04.2025

Рисунок 3.44 – Сформувати звіт про дохід за період

Preview - Tabular information

Print Printer setup Scale Previous Next Stop Close

Дохід за період з 09.03.2025 по 09.06.2025

№	ПІБ клієнта	Назва секції	Вартість зан	Дата придбання
24	Шипко Андрій Федорович	теніс настільний	600	01.04.2025
25	Глазунов Сергій Миколайович	теніс настільний	600	01.04.2025
26	Балицький Євген Віталійович	спортивне орієнтування	300	01.04.2025
27	Марговицький Артур Володимирович	міні-футбол (чоловіки)	600	31.03.2025
28	Павлов Костянтин Юрійович	боротьба спортивна	690	25.04.2025

Page 1 of 1

Рисунок 3.45 – Друк звіту про дохід за період

Звіт

Звіт по клієнтам

Друк

№	ПІБ клієнта	Назва секції	Вартість зан	Дата придбання
22	Васильєв Александр Андреевич	рукопашний бій	630	11.02.2025
23	Глазунов Сергій Миколайович	спортивна аеробіка	690	19.02.2025
24	Шипко Андрій Федорович	теніс настільний	600	01.04.2025
25	Глазунов Сергій Миколайович	теніс настільний	600	01.04.2025
26	Балицький Євген Віталійович	спортивне орієнтування	300	01.04.2025
27	Марговицький Артур Володимирович	міні-футбол (чоловіки)	600	31.03.2025
28	Павлов Костянтин Юрійович	боротьба спортивна	690	25.04.2025
30	Марговицький Артур Володимирович	гиревий спорт	600	09.06.2025
31	Бандуров Володимир Володимирович	боротьба спортивна	690	09.06.2025

Рисунок 3.46 – Звіт по клієнтам

Preview - Tabular information

Print Printer setup Scale Previous Next Stop Close

ЗВІТ ПО КЛІЄНТАМ

№	ПІБ клієнта	Назва секції	Вартість зан	Дата придбання
22	Васильєв Александр Андреевич	рукопашний бій	630	11.02.2025
23	Глазунов Сергій Миколайович	спортивна аеробіка	690	19.02.2025
24	Шипко Андрій Федорович	теніс настільний	600	01.04.2025
25	Глазунов Сергій Миколайович	теніс настільний	600	01.04.2025
26	Балицький Євген Віталійович	спортивне орієнтування	300	01.04.2025
27	Марговицький Артур Володимирович	міні-футбол (чоловіки)	600	31.03.2025
28	Павлов Костянтин Юрійович	боротьба спортивна	690	25.04.2025
30	Марговицький Артур Володимирович	гиревий спорт	600	09.06.2025
31	Бандуров Володимир Володимирович	боротьба спортивна	690	09.06.2025

Page 1 of 1

Рисунок 3.47 – Друк звіту по клієнтам

ВИСНОВКИ

Метою роботи була розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі, призначеної для надання оперативної інформації для обліку і подальшого її аналізу та зниження обсягів паперового документообігу.

Для її реалізації було зроблено:

Зроблено аналіз інформаційних систем, вивчені їх класифікація, архітектура та створення.

Досліджено інформаційні процеси при контролі абонементів спортивно-оздоровчого комплексу.

За допомогою технологій HTML, CSS, JavaScript для створення клієнтської частини та технологій Node.js, AJAX, Firebird Database для створення серверної частини розроблена інформаційна система для контролю абонементів спортивно-оздоровчого комплексу, призначена для надання оперативної інформації для обліку та подальшого аналізу.

Розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі включає:

- створення бази даних для інформаційної системи;
- створення форм для надання нормативно-довідникової інформації (довідники клієнтів, тренерів та спортивних секцій);
- створення вхідних документів (створення абоненту з можливістю його друку, прайс-лист послуг);
- створення вихідних документів (дохід за період, звіт по клієнтам з можливістю друку цих документів).

Розробка інформаційної системи буде сприяти підвищенню продуктивності праці співробітників спортивного комплексу, оскільки надасть можливість оперативно збирати та аналізувати інформацію.

ПЕРЕЛІК ПОСИЛАНЬ

1. Коннолли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. М.: Вильямс, 2015. – 1440 с.
2. Крёнке Д. Теория и практика построения баз данных. – СПб: Питер, 2016. – 800с.
3. Дунаев В.В. Базы данных. Язык SQL для студента. – СПб: БХВ- Петербург, 2017 – 312с.
4. Марков А.С., Лисовский К.Ю. Базы данных. Введение в теорию и методологию: Учебник. – М.: Финансы и статистика, 2016. – 512 с.
5. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. Проектирование информационных систем. – М.: Интернет-Ун-т Информ технологий, 2015. – 304 с.
6. Блинков Ю.А. Проектирование информационных систем. – Саратов: Саратовский государственный университет, 2016. – 377 с.
7. Антоненко В.М. Сучасні інформаційні системи і технології: управління знаннями : навч. посібник / В. М. Антоненко, С. Д. Мамченко, Ю. В. Рогушина. – Ірпінь: Нац. університет ДПС України, 2016. – 212 с.
8. Воронін А.М., Зіатдінов Ю.К., Климова А.С. Інформаційні системи прийняття рішень: навчальний посібник. – К.: НАУ-друк, 2019. – 136с.
9. Галузинський Г.П., Денісова О.О., Писаревська Т.А. Інформаційні системи у бізнесі. Практикум для індивідуальної роботи: навч.-метод. посіб. для самост. вивч. дисципліни. – К. : КНЕУ, 2018. – 524с.
10. Годун В.М., Орленко Н.С., Сендзюк М.А. Інформаційні системи і технології в статистиці: навч. посіб. – К.: КНЕУ, 2016. – 267 с.
11. Петров В.Н. Информационные системы. – СПб: Питер, 2012. – 688 с.
12. Семенов М.И., Трубилин И.Т., Лойко В.И., Барановская Т.П. Автоматизированные информационные технологии в экономике: Учебник. – М: Финансы и статистика, 2013. – 627 с.

13. Терещенко Л.О. Інформаційні системи і технології в обліку: Навчальний посібник. – К.: КНЕУ, 2014. – 187 с.
14. Хомоненко А.Д., Цыганков В.М. Базы данных: Учебник для высших учебных заведений. – СПб: Корона, 2014. – 736 с.
15. Хансен Г., Хансен Дж. Базы данных: разработка и управление. – М.: ЗАО «Издательство БИНОМ», 2002. – 704 с.
16. Роб П., Коронел К. Системы баз данных: проектирование, реализация и управление. – СПб.: БХВ-Петербург, 2014. – 1040 с.
17. Кузин А.В., Левонисова С.В. Базы данных: учеб. Пособие для студ. высш. учеб. заведений. – М.: Изд. центр «Академия», 2018. – 320с.
18. Кузнецов С. Д. Основы баз данных: учебное пособие. – М.: Интернет-Университет Информационных Технологий; БИНОМ. Лаборатория знаний, 2017. – 484 с.
19. Завадський І.О. Основи баз даних: навч. посіб. – К.: Видавець І.О. Завадський, 2016. – 192 с.
20. Андон Ф., Резниченко В. Язык запросов SQL. Учебный курс. – СПб: Питер; Киев: Издательская группа BHV, 2016. – 416 с.
21. Бен-Ган И. MSSQL Server 2008. Основы T-SQL. – СПб: БХВ-Питер, 2019. – 430с.
22. Анісімов А.В., Кулябко П.П. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. – Київ: КНУ ім. Т. Шевченка, 2017. – 110 с.
23. Дейт К.Дж. Введение в системы баз данных. – К.: Диалектика, 2014. – 784 с.
24. Бойков В.А., Савинков В.М. Проектирование баз данных информационных систем. – М. Мир, 2007. – 252 с.
25. Хансен Г., Хансен Дж. Базы данных: разработка и управление. – М.: БИНОМ, 2012. – 704 с.
26. Симонов Д., Винкенуг П., Филиппов Д., Еманов Д., Карпейкин А., Кузьменко Д., Ковязин А. Руководство по языку SQL СУБД Firebird 2.5: Firebird 2.5.9. – 569 с. – [Електроний ресурс] – URL:

https://firebirdsql.org/file/documentation/reference_manuals/Firebird_Language_Reference_RUS.pdf.

27. Бондарь А. InterBase и Firebird. Практическое руководство для умных пользователей и начинающих разработчиков. – СПб: BHV, 2017. – 592 с.

28. Фролов А.В., Фролов Г.В. Создание Web-приложений: Практическое руководство. – М.: Издательско-торговый дом "Русская редакция", 2011. – 1040 с.

29. Федорчук А. Как создаются Web-сайты. Краткий курс. – СПб: Питер, 2010. – 224 с.

30. Нельсен Якоб. Веб дизайн. – СПб.: Питер, 2013. – 504 с.

31. Пауэлл Т.А. Полное руководство по HTML. – Мн.: ООО "Попурри", 2011. – 912 с.

32. Мальчук Е.В. HTML и CSS. Самоучитель. М.: Издательский дом "Вильямс". – 2018. – 416 с.

33. Дубаков М.А. Веб-мастеринг средствами CSS. – СПб.: БХВ-Петербург, 2012. – 544 с.

34. Кастро Э. Создание Web-страниц с помощью HTML. М.: НТ Пресс, 2015. – 144 с.

35. Росс В.С. Створення сайтів: HTML, CSS, PHP, MySQL: підручник, ч. 1 – М.: МГДД(Ю)Т, 2020. – 107 с.

36. Флэнаган Д. JavaScript: Посібник. – СПб: Символ Плюс, 2008. – 992 с.

37. Підручник learn Javascript [Електронний ресурс]. – URL: <https://learn.javascript.ru>

38. Гаевский А.Ю. 100% самоучитель. Создание Web-страниц и Web-сайтов. HTML и JavaScript / А.Ю. Гаевский, В.А. Романовский. – М.: Наука, 2020. – 464 с.

39. Дронов, В. JavaScript в Web-дизайне / В. Дронов. – М.: СПб: БХВ, 2014. – 880 с.

40. Дунаев Вадим. HTML, скрипты и стили. Спб.: БХВ- Петербург, 2008. – 1024 с.

41. Исси Коэн Л. Полный справочник по HTML, CSS и JavaScript / Лазаро Исси Коэн, Джозеф Исси Коэн. – М.: ЭКОМ Паблишерз, 2019. – 938 с.

42. Node.js Documentation – Офіційна документація Node.js. URL:
<https://nodejs.org/en/docs/>

43. Прохоренок Н., Дронов В. JavaScript и Node.js для веб-разработчиков. – СПб: БХВ, 2022. – 768 с.

ФРАГМЕНТЫ ПРОГРАМНОГО КОДУ

```

/*****
/*** Generated by IBExpert 28.04.2025 07:08:35 ***/
/*****/
SET SQL DIALECT 3;
SET NAMES NONE;
CREATE DATABASE 'D:\Gym\db\GYM.GDB'
USER 'SYSDBA' PASSWORD 'masterkey'
PAGE_SIZE 16384
DEFAULT CHARACTER SET NONE COLLATION NONE;
/*****/
/*** Stored procedures ***/
/*****/
SET TERM ^ ;
CREATE PROCEDURE ABON_DEL (
KOD_ABON INTEGER /* TYPE OF COLUMN ABON.KOD_ABON */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE ABON_INS (
KOD_ABON INTEGER /* TYPE OF COLUMN ABON.KOD_ABON */,
KOD_KL INTEGER /* TYPE OF COLUMN ABON.KOD_KL */,
KOD_SPSEC INTEGER /* TYPE OF COLUMN ABON.KOD_SPSEC */,
DATA_POK DATE /* TYPE OF COLUMN ABON.DATA_POK */,
VID_1 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_1 */,
VID_2 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_2 */,
VID_3 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_3 */,
VID_4 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_4 */,
VID_5 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_5 */,
VID_6 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_6 */,
VID_7 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_7 */,
VID_8 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_8 */,
VID_9 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_9 */,
VID_10 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_10 */,
VID_11 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_11 */,
VID_12 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_12 */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE ABON_UPD (
KOD_ABON INTEGER /* TYPE OF COLUMN ABON.KOD_ABON */,
KOD_KL INTEGER /* TYPE OF COLUMN ABON.KOD_KL */,
KOD_SPSEC INTEGER /* TYPE OF COLUMN ABON.KOD_SPSEC */,
DATA_POK DATE /* TYPE OF COLUMN ABON.DATA_POK */,
VID_1 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_1 */,
78
VID_2 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_2 */,
VID_3 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_3 */,
VID_4 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_4 */,
VID_5 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_5 */,

```

```

VID_6 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_6 */,
VID_7 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_7 */,
VID_8 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_8 */,
VID_9 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_9 */,
VID_10 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_10 */,
VID_11 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_11 */,
VID_12 VARCHAR(1) /* TYPE OF COLUMN ABON.VID_12 */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE KL_DEL (
KOD_KL INTEGER /* TYPE OF COLUMN KL.KOD_KL */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE KL_INS (
KOD_KL INTEGER /* TYPE OF COLUMN KL.KOD_KL */,
FIO_KL VARCHAR(50) /* TYPE OF COLUMN KL.FIO_KL */,
SERPAS_KL VARCHAR(2) /* TYPE OF COLUMN KL.SERPAS_KL */,
NOMPAS_KL INTEGER /* TYPE OF COLUMN KL.NOMPAS_KL */,
TEL_KL VARCHAR(10) /* TYPE OF COLUMN KL.TEL_KL */,
EMAIL_KL VARCHAR(20) /* TYPE OF COLUMN KL.EMAIL_KL */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE KL_UPD (
KOD_KL INTEGER /* TYPE OF COLUMN KL.KOD_KL */,
FIO_KL VARCHAR(50) /* TYPE OF COLUMN KL.FIO_KL */,
SERPAS_KL VARCHAR(2) /* TYPE OF COLUMN KL.SERPAS_KL */,
NOMPAS_KL INTEGER /* TYPE OF COLUMN KL.NOMPAS_KL */,
TEL_KL VARCHAR(10) /* TYPE OF COLUMN KL.TEL_KL */,
EMAIL_KL VARCHAR(20) /* TYPE OF COLUMN KL.EMAIL_KL */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE SPSEC_DEL (
KOD_SPSEC INTEGER /* TYPE OF COLUMN SPSEC.KOD_SPSEC */)
AS
BEGIN
EXIT;
END^
79
CREATE PROCEDURE SPSEC_INS (
KOD_SPSEC INTEGER /* TYPE OF COLUMN SPSEC.KOD_SPSEC */,
NAME_SPSEC VARCHAR(100) /* TYPE OF COLUMN SPSEC.NAME_SPSEC */,
DNI VARCHAR(50) /* TYPE OF COLUMN SPSEC.DNI */,
STOIM NUMERIC(15,2) /* TYPE OF COLUMN SPSEC.STOIM */,
KOD_TR INTEGER /* TYPE OF COLUMN SPSEC.KOD_TR */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE SPSEC_UPD (

```

```

KOD_SPSEC INTEGER /* TYPE OF COLUMN SPSEC.KOD_SPSEC */,
NAME_SPSEC VARCHAR(100) /* TYPE OF COLUMN SPSEC.NAME_SPSEC */,
DNI VARCHAR(50) /* TYPE OF COLUMN SPSEC.DNI */,
STOIM NUMERIC(15,2) /* TYPE OF COLUMN SPSEC.STOIM */,
KOD_TR INTEGER /* TYPE OF COLUMN SPSEC.KOD_TR */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE TR_DEL (
KOD_TR INTEGER /* TYPE OF COLUMN TR.KOD_TR */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE TR_INS (
KOD_TR INTEGER /* TYPE OF COLUMN TR.KOD_TR */,
FIO_TR VARCHAR(50) /* TYPE OF COLUMN TR.FIO_TR */,
TEL_TR VARCHAR(10) /* TYPE OF COLUMN TR.TEL_TR */,
EMAIL_TR VARCHAR(20) /* TYPE OF COLUMN TR.EMAIL_TR */,
ZVAN_TR VARCHAR(250) /* TYPE OF COLUMN TR.ZVAN_TR */,
DOST_TR VARCHAR(250) /* TYPE OF COLUMN TR.DOST_TR */)
AS
BEGIN
EXIT;
END^
CREATE PROCEDURE TR_UPD (
KOD_TR INTEGER /* TYPE OF COLUMN TR.KOD_TR */,
FIO_TR VARCHAR(50) /* TYPE OF COLUMN TR.FIO_TR */,
TEL_TR VARCHAR(10) /* TYPE OF COLUMN TR.TEL_TR */,
EMAIL_TR VARCHAR(20) /* TYPE OF COLUMN TR.EMAIL_TR */,
ZVAN_TR VARCHAR(250) /* TYPE OF COLUMN TR.ZVAN_TR */,
DOST_TR VARCHAR(250) /* TYPE OF COLUMN TR.DOST_TR */)
AS
BEGIN
EXIT;
END^
SET TERM ; ^
80
/*****
/
*** Tables ***/
/
*****/
CREATE TABLE ABON (
KOD_ABON INTEGER NOT NULL,
KOD_KL INTEGER NOT NULL,
KOD_SPSEC INTEGER NOT NULL,
DATA_POK DATE,
VID_1 VARCHAR(1),
VID_2 VARCHAR(1),
VID_3 VARCHAR(1),
VID_4 VARCHAR(1),
VID_5 VARCHAR(1),
VID_6 VARCHAR(1),
VID_7 VARCHAR(1),
VID_8 VARCHAR(1),
VID_9 VARCHAR(1),
VID_10 VARCHAR(1),

```

```

VID_11 VARCHAR(1),
VID_12 VARCHAR(1)
);
CREATE TABLE KL (
KOD_KL INTEGER NOT NULL,
FIO_KL VARCHAR(50),
SERPAS_KL VARCHAR(2),
NOMPAS_KL INTEGER,
TEL_KL VARCHAR(10),
EMAIL_KL VARCHAR(20)
);
CREATE TABLE SPSEC (
KOD_SPSEC INTEGER NOT NULL,
NAME_SPSEC VARCHAR(100),
DNI VARCHAR(50),
STOIM NUMERIC(15,2),
KOD_TR INTEGER NOT NULL
);
CREATE TABLE TR (
KOD_TR INTEGER NOT NULL,
FIO_TR VARCHAR(50),
TEL_TR VARCHAR(10),
EMAIL_TR VARCHAR(20),
ZVAN_TR VARCHAR(250),
DOST_TR VARCHAR(250)
);
81
/*****
/**** Primary keys ****/
/*****
ALTER TABLE ABON ADD CONSTRAINT PK_ABON PRIMARY KEY (KOD_ABON);
ALTER TABLE KL ADD CONSTRAINT PK_KL PRIMARY KEY (KOD_KL);
ALTER TABLE SPSEC ADD CONSTRAINT PK_SPSEC PRIMARY KEY (KOD_SPSEC);
ALTER TABLE TR ADD CONSTRAINT PK_TR PRIMARY KEY (KOD_TR);
/*****
/**** Foreign keys ****/
/*****
ALTER TABLE ABON ADD CONSTRAINT FK_ABON_1 FOREIGN KEY (KOD_KL)
REFERENCES KL (KOD_KL);
ALTER TABLE SPSEC ADD CONSTRAINT FK_SPSEC_1 FOREIGN KEY (KOD_TR)
REFERENCES TR (KOD_TR);
/*****
/**** Stored procedures ****/
/*****
SET TERM ^ ;
ALTER PROCEDURE ABON_DEL (
KOD_ABON TYPE OF COLUMN ABON.KOD_ABON)
AS
begin
delete from abon
where (kod_abon = :kod_abon);
end^
ALTER PROCEDURE ABON_INS (
KOD_ABON TYPE OF COLUMN ABON.KOD_ABON,
KOD_KL TYPE OF COLUMN ABON.KOD_KL,
KOD_SPSEC TYPE OF COLUMN ABON.KOD_SPSEC,
DATA_POK TYPE OF COLUMN ABON.DATA_POK,

```

```

VID_1 TYPE OF COLUMN ABON.VID_1,
VID_2 TYPE OF COLUMN ABON.VID_2,
VID_3 TYPE OF COLUMN ABON.VID_3,
VID_4 TYPE OF COLUMN ABON.VID_4,
VID_5 TYPE OF COLUMN ABON.VID_5,
VID_6 TYPE OF COLUMN ABON.VID_6,
VID_7 TYPE OF COLUMN ABON.VID_7,
VID_8 TYPE OF COLUMN ABON.VID_8,
VID_9 TYPE OF COLUMN ABON.VID_9,
VID_10 TYPE OF COLUMN ABON.VID_10,
VID_11 TYPE OF COLUMN ABON.VID_11,
VID_12 TYPE OF COLUMN ABON.VID_12)
AS
begin
insert into abon (
kod_abon, kod_kl, kod_spsec, data_pok,
82
vid_1, vid_2, vid_3, vid_4, vid_5, vid_6, vid_7, vid_8, vid_9, vid_10, vid_11, vid_12)
values (
:kod_abon, :kod_kl, :kod_spsec, :data_pok,
:vid_1, :vid_2, :vid_3, :vid_4, :vid_5, :vid_6, :vid_7, :vid_8, :vid_9, :vid_10, :vid_11, :vid_12);
end^
ALTER PROCEDURE ABON_UPD (
KOD_ABON TYPE OF COLUMN ABON.KOD_ABON,
KOD_KL TYPE OF COLUMN ABON.KOD_KL,
KOD_SPSEC TYPE OF COLUMN ABON.KOD_SPSEC,
DATA_POK TYPE OF COLUMN ABON.DATA_POK,
VID_1 TYPE OF COLUMN ABON.VID_1,
VID_2 TYPE OF COLUMN ABON.VID_2,
VID_3 TYPE OF COLUMN ABON.VID_3,
VID_4 TYPE OF COLUMN ABON.VID_4,
VID_5 TYPE OF COLUMN ABON.VID_5,
VID_6 TYPE OF COLUMN ABON.VID_6,
VID_7 TYPE OF COLUMN ABON.VID_7,
VID_8 TYPE OF COLUMN ABON.VID_8,
VID_9 TYPE OF COLUMN ABON.VID_9,
VID_10 TYPE OF COLUMN ABON.VID_10,
VID_11 TYPE OF COLUMN ABON.VID_11,
VID_12 TYPE OF COLUMN ABON.VID_12)
AS
begin
update abon
set kod_kl = :kod_kl,
kod_spsec = :kod_spsec,
data_pok = :data_pok,
vid_1 = :vid_1,
vid_2 = :vid_2,
vid_3 = :vid_3,
vid_4 = :vid_4,
vid_5 = :vid_5,
vid_6 = :vid_6,
vid_7 = :vid_7,
vid_8 = :vid_8,
vid_9 = :vid_9,
vid_10 = :vid_10,
vid_11 = :vid_11,
vid_12 = :vid_12

```

```

where (kod_abon = :kod_abon);
end^
ALTER PROCEDURE KL_DEL (
KOD_KL TYPE OF COLUMN KL.KOD_KL)
AS
begin
delete from kl
where (kod_kl = :kod_kl);
end^
83
ALTER PROCEDURE KL_INS (
KOD_KL TYPE OF COLUMN KL.KOD_KL,
FIO_KL TYPE OF COLUMN KL.FIO_KL,
SERPAS_KL TYPE OF COLUMN KL.SERPAS_KL,
NOMPAS_KL TYPE OF COLUMN KL.NOMPAS_KL,
TEL_KL TYPE OF COLUMN KL.TEL_KL,
EMAIL_KL TYPE OF COLUMN KL.EMAIL_KL)
AS
begin
insert into kl ( kod_kl, fio_kl, serpas_kl, nompas_kl, tel_kl, email_kl)
values ( :kod_kl, :fio_kl, :serpas_kl, :nompas_kl, :tel_kl, :email_kl);
end^
ALTER PROCEDURE KL_UPD (
KOD_KL TYPE OF COLUMN KL.KOD_KL,
FIO_KL TYPE OF COLUMN KL.FIO_KL,
SERPAS_KL TYPE OF COLUMN KL.SERPAS_KL,
NOMPAS_KL TYPE OF COLUMN KL.NOMPAS_KL,
TEL_KL TYPE OF COLUMN KL.TEL_KL,
EMAIL_KL TYPE OF COLUMN KL.EMAIL_KL)
AS
begin
update kl
set fio_kl = :fio_kl,
serpas_kl = :serpas_kl,
nompas_kl = :nompas_kl,
tel_kl = :tel_kl,
email_kl = :email_kl
where (kod_kl = :kod_kl);
end^
ALTER PROCEDURE SPSEC_DEL (
KOD_SPSEC TYPE OF COLUMN SPSEC.KOD_SPSEC)
AS
begin
delete from spsec
where (kod_spsec = :kod_spsec);
end^
ALTER PROCEDURE SPSEC_INS (
KOD_SPSEC TYPE OF COLUMN SPSEC.KOD_SPSEC,
NAME_SPSEC TYPE OF COLUMN SPSEC.NAME_SPSEC,
DNI TYPE OF COLUMN SPSEC.DNI,
STOIM TYPE OF COLUMN SPSEC.STOIM,
KOD_TR TYPE OF COLUMN SPSEC.KOD_TR)
AS
begin
insert into spsec ( kod_spsec, name_spsec, dni, stoim, kod_tr)
values ( :kod_spsec, :name_spsec, :dni, :stoim, :kod_tr);
end^

```

```

ALTER PROCEDURE SPSEC_UPD (
KOD_SPSEC TYPE OF COLUMN SPSEC.KOD_SPSEC,
NAME_SPSEC TYPE OF COLUMN SPSEC.NAME_SPSEC,
DNI TYPE OF COLUMN SPSEC.DNI,
STOIM TYPE OF COLUMN SPSEC.STOIM,
84
KOD_TR TYPE OF COLUMN SPSEC.KOD_TR)
AS
begin
update spsec
set name_spsec = :name_spsec,
dni = :dni,
stoim = :stoim,
kod_tr = :kod_tr
where (kod_spsec = :kod_spsec);
end^
ALTER PROCEDURE TR_DEL (
KOD_TR TYPE OF COLUMN TR.KOD_TR)
AS
begin
delete from tr
where (kod_tr = :kod_tr);
end^
ALTER PROCEDURE TR_INS (
KOD_TR TYPE OF COLUMN TR.KOD_TR,
FIO_TR TYPE OF COLUMN TR.FIO_TR,
TEL_TR TYPE OF COLUMN TR.TEL_TR,
EMAIL_TR TYPE OF COLUMN TR.EMAIL_TR,
ZVAN_TR TYPE OF COLUMN TR.ZVAN_TR,
DOST_TR TYPE OF COLUMN TR.DOST_TR)
AS
begin
insert into tr ( kod_tr, fio_tr, tel_tr, email_tr, zvan_tr, dost_tr)
values ( :kod_tr, :fio_tr, :tel_tr, :email_tr, :zvan_tr, :dost_tr);
end^
ALTER PROCEDURE TR_UPD (
KOD_TR TYPE OF COLUMN TR.KOD_TR,
FIO_TR TYPE OF COLUMN TR.FIO_TR,
TEL_TR TYPE OF COLUMN TR.TEL_TR,
EMAIL_TR TYPE OF COLUMN TR.EMAIL_TR,
ZVAN_TR TYPE OF COLUMN TR.ZVAN_TR,
DOST_TR TYPE OF COLUMN TR.DOST_TR)
AS
begin
update tr
set fio_tr = :fio_tr,
tel_tr = :tel_tr,
email_tr = :email_tr,
zvan_tr = :zvan_tr,
dost_tr = :dost_tr
where (kod_tr = :kod_tr);
end^
SET TERM ; ^

```

Файл main.css

```
body {
  margin: 0;
  padding: 0;
  font-size: 18px;
  line-height: 24px;
  font-family: "PT Sans", "Arial", sans-serif;
  min-width: 1040px;
  color: #404040;
  background-color: white;
}

.container {
  width: 80%;
  margin: 0 auto;
  max-width: 1400px;
}

.header {
  min-height: 350px;
  padding: 60px 0 0 0;
  margin: 0;
  color: white;
  background-color: #363636;
}

.flexbox-header {
  position: relative;
  display: flex;
  justify-content: space-between;
}

.header .list-menu {
  margin: 0;
  display: flex;
  height: 30px;
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
  align-self: flex-end;
  width: 800px;
  list-style: none;
}

.header .list-menu li:first-child {
  margin-right: 10px;
  position: relative;
}

.header .list-menu li:first-child p {
  margin-left: -40px;
  color: white;
}

.header .list-menu li:first-child p::after {
  content: "";
  display: block;
  position: absolute;
  top: 27px;
  left: 90px;
  width: 0;
```

```
height: 0;
border: 8px solid white;
border-bottom: none;
border-right-color: transparent;
border-left-color: transparent;
}
```

```
.header .list-menu li:first-child ul.numbers {
/*display: none;*/
opacity: 0;
position: absolute;
margin: -10px 0 0 -40px;
padding: 0 0 0 5px;
list-style: none;
z-index: 100;
transition-property: opacity;
transition-duration: 0.8s;
}
```

```
.header .list-menu li:hover ul.numbers {
/*display: block;*/
opacity: 1;
transition-property: opacity;
transition-duration: 0.5s;
}
```

```
.header .list-menu li:first-child ul.numbers li {
width: 131px;
border-bottom: 1px solid white;
}
.header .list-menu li a {
color: white;
text-decoration: none;
}
```

```
.header .list-menu li a:hover {
text-decoration: underline;
}
```

```
.header .list-menu li a:active {
opacity: 0.7;
}
```

```
.header .flexbox-header div:last-of-type {
height: 53px;
align-self: flex-end;
margin-bottom: -40px;
position: relative;
padding-bottom: 10px;
}
```

```
.header .center {
height: 70px;
width: 100px;
}
```

```
.header .center::before {
content: attr(data-amount) "";
font-size: 23px;
line-height: 23px;
position: absolute;
display: block;
```

```
color: #404040;
width: 30px;
height: 23px;
top: 2px;
right: 9px;
vertical-align: middle;
text-align: center;
font-weight: bolder;
}
```

```
.header .center::after {
position: absolute;
content: attr(data-total) "";
font-size: 16px;
line-height: 20px;
display: block;
width: 100%;
height: 20px;
bottom: 0;
left: 0;
vertical-align: middle;
text-align: center;
font-weight: bolder;
color: #a38b70;
}
```

```
.header .slogan {
margin-top: 90px;
font-family: "Georgia", serif;
font-size: 45px;
line-height: 55px;
font-weight: bold;
text-align: center;
}
```

```
.section-title {
font-family: "Georgia", serif;
font-size: 45px;
line-height: 55px;
font-weight: bold;
text-align: center;
color: #a38b70;
}
```

```
.section-title-mini {
font-size: 30px;
line-height: 35px;
}
```

```
.btn {
border-radius: 2px;
font: inherit;
word-wrap: break-word;
width: 140px;
text-align: center;
display: inline-block;
vertical-align: top;
padding: 10px;
border: 2px solid #a38b70;
color: #a38b70;
text-decoration: none;
text-transform: uppercase;
}
```

```

}

.btn:hover {
  color: #8d745a;
  border-color: #8d745a;
}

.btn:active {
  color: rgba(141, 116, 90, 0.6);
  border-color: rgba(141, 116, 90, 0.6);
}
/*
.btn-reviews {
  color: #404040;
  border-color: #404040;
}

.btn-reviews:hover {
  color: rgba(64, 64, 64, 0.7);
  border-color: rgba(64, 64, 64, 0.7);
}

.btn-reviews:active {
  color: rgba(64, 64, 64, 0.5);
  border-color: rgba(64, 64, 64, 0.5);
}
*/
.some-information {
  padding: 30px 0 50px 0;
  text-align: center;
}

.features {
  padding: 70px 0 75px 0;
  background-color: #f2f2f2;
}

.features ul.features-flexbox {
  margin: 0;
  padding: 0;
  display: flex;
  flex-direction: row;
  justify-content: space-around;
  align-items: center;
  list-style: none;
}

.features ul li {
  width: 100%;
}

.features ul li div.feature-time {
  background: url("../images/feature-time.png") no-repeat top center;
}

.features ul li div.feature-care {
  background: url("../images/feature-care.png") no-repeat top center;
}

.features ul li div.feature-big {
  background: url("../images/feature-big.png") no-repeat top center;
}

```

```
.features ul li div.feature-content {  
  margin: 0 auto 0 auto;  
  padding-top: 140px;  
  width: 300px;  
  text-align: center;  
  background-size: 150px auto;  
}
```

```
.features ul li:nth-child(2) {  
  border-left: 2px solid #a38b70;  
  border-right: 2px solid #a38b70;  
}
```

```
.features ul li .feature-title {  
  font-family: "Georgia", serif;  
  font-size: 30px;  
  line-height: 30px;  
  font-weight: bold;  
  color: #a38b70;  
}
```

```
.features ul li p {  
  font-size: 16px;  
}
```

```
.reviews {  
  padding: 70px 0 90px 0;  
}
```

```
.reviews a.btn {  
  float: right;  
  margin-top: -23px;  
}
```

```
.reviews .reviews-list {  
  margin-top: 80px;  
  display: flex;  
  flex-direction: row;  
  justify-content: space-between;  
  align-items: center;  
}
```

```
.reviews .review-item {  
  width: 45%;  
  font-size: 16px;  
  border-left: 3px solid #a38b70;  
  padding: 0 0 0 20px;  
  margin: 0;  
}
```

```
.reviews .reviews-list p {  
  margin-top: 3px;  
  word-wrap: break-word;  
}
```

```
.reviews .review-author {  
  font-weight: bold;  
  font-style: italic;  
}
```

```
.workers {
```

```
padding: 50px 0 70px 0;
background-color: #f2f2f2;
}
```

```
.workers .photoes {
margin: 0;
display: flex;
flex-direction: row;
justify-content: space-around;
align-content: center;
align-items: center;
flex-wrap: wrap;
}
```

```
.workers .photo {
/*flex-grow: 1;*/
width: 330px;
margin-bottom: 15px;
}
```

```
.workers .photo img {
width: 95%;
height: auto;
margin: 2.5%;
margin-bottom: 0;
}
```

```
.workers .photo p {
text-align: center;
}
```

```
.footer {
margin: 0;
padding: 45px 0 50px;
background-color: #363636;
color: #c3c3c3;
}
```

```
.footer .footer-top {
display: flex;
width: 100%;
justify-content: space-between;
align-items: center;
margin-bottom: 45px;
}
```

```
.footer .footer-top img {
margin-left: 30px;
}
```

```
.footer .footer-top a {
margin-right: 20px;
}
```

```
.footer .footer-middle {
display: flex;
width: 100%;
justify-content: space-around;
align-items: flex-start;
padding-top: 50px;
margin-bottom: 50px;
}
```

```
.footer .footer-middle,  
.footer .footer-bottom {  
  border-top: 1px solid #5e5e5e;  
}  
  
.footer .nav-list ul {  
  list-style: none;  
  padding: 0;  
  margin: 0;  
}  
  
.footer .nav-list ul li {  
  margin-bottom: 10px;  
}  
  
.footer .nav-list ul li p {  
  margin: 0;  
  font-size: 16px;  
  color: #c3c3c3;  
}  
  
.footer .nav-list ul li a {  
  font-size: 16px;  
  color: #c3c3c3;  
  text-decoration: none;  
}  
  
.footer .socials {  
  padding-left: 20px;  
}  
  
.footer .socials b {  
  font-family: "Georgia", serif;  
  font-size: 24px;  
  color: #a38b70;  
  margin-right: 40px;  
}  
  
.footer .socials a:last-child {  
  background-image: url("../images/inst.png");  
}  
  
.footer .socials a:hover {  
  border-color: rgba(255, 255, 255, 0.3);  
}  
  
.footer .socials a:active {  
  color: #ffffff;  
  opacity: 0.1;  
}  
  
.footer .copyright {  
  padding-right: 20px;  
}  
.footer .copyright p {  
  color: #ffffff;  
  text-transform: uppercase; }
```

Файл order_script.js

/*для категорий*/

```
var check_remove_add = function(selector) {
    var elem = document.querySelector('.active');
    if (elem) {
        elem.classList.remove('active');
    }
    document.querySelector(selector).classList.add('active');
}
```

/*для загрузки информации*/

```
var downloadData = function() {
    var supermarkets = document.querySelectorAll('.center');
    centers.forEach(function(center) {
        var requestURL = center.dataset.url;
        var request = new XMLHttpRequest();
        request.open('GET', requestURL);
        request.responseType = '';
        request.send();
        request.onload = function() {
            var klients = request.response;
            center.dataset.klients = klients;
        };
    });
}
```

```
document.querySelector('.category:nth-child(1)').onclick = function() {check_remove_add('.category:nth-child(1));}
document.querySelector('.category:nth-child(2)').onclick = function() {check_remove_add('.category:nth-child(2));}
document.querySelector('.category:nth-child(3)').onclick = function() {check_remove_add('.category:nth-child(3));}
document.querySelector('.category:nth-child(4)').onclick = function() {check_remove_add('.category:nth-child(4));}
document.querySelector('.category:nth-child(5)').onclick = function() {check_remove_add('.category:nth-child(5));}
document.querySelector('.category:nth-child(6)').onclick = function() {check_remove_add('.category:nth-child(6));}
document.querySelector('.category:nth-child(7)').onclick = function() {check_remove_add('.category:nth-child(7));}
document.querySelector('.category:nth-child(8)').onclick = function() {check_remove_add('.category:nth-child(8));}
document.querySelector('.category:nth-child(9)').onclick = function() {check_remove_add('.category:nth-child(9));}
document.querySelector('.category:nth-child(10)').onclick = function() {check_remove_add('.category:nth-child(10));}
```

/*для магазинов*/

```
var check_remove_add_sup = function(selector) {
    var elem_super = document.querySelector('.active-super');
    if (elem_super) {
        elem_super.classList.remove('active-super');
    }
    document.querySelector(selector).classList.add('active-super');
}
```

```
var createElement = function (tagName, className) {
    var element = document.createElement('tagName');
    if (className) {
        element.classList.add(className);
    };
    return element;
}
```

```
var createKlients = function() {
    var product = createElement("li", "product");
```

```

        product.insertAdjacentHTML("beforeEnd", "<div class='product-around-image' data-price=" + price + "><p class='product-
name'></p></div>");
        product.insertAdjacentHTML("beforeEnd", "<input type='number' value='1' min='1' max='10' name='amount'>");
        product.insertAdjacentHTML("beforeEnd", "<input type='button' value='Додати' name='to-center'>");
        product.insertAdjacentHTML("beforeEnd", "<output name='total-price'></output>");
        return product;
    }

    var getDataCreate = function() {
        clear();
        var preData = document.querySelector('.active .active-super').dataset.klients;
        var dataArray = preData.split(' ');
        var listContain = document.createDocumentFragment();
        for (var i = 0; i < dataArray.length; i++) {
            var listElement = createKlients();
            listContain.appendChild(listElement);
        };
        var parent = document.querySelector('.klients');
        parent.style.display = 'flex';
        parent.appendChild(listContain);
        valueElements(dataArray);
    }

    var valueElements = function(dataArray) {
        for (var i = 0; i < dataArray.length; i++) {
            var object = JSON.parse(dataArray[i]);
            var num = i+1;
            if (object.specialOffer) {
                var li = document.querySelector('.product:nth-child('+num+')');
                li.classList.add("special-offer");
            }
            if (object.available) {
                var li = document.querySelector('.product:nth-child('+num+')');
                li.classList.add("not-available");
                var amount = document.querySelector('.product:nth-child('+num+') input[name*="amount"]');
                amount.setAttribute('readonly', '');
                var button = document.querySelector('.product:nth-child('+num+') input[name*="to-center"]');
                button.setAttribute('readonly', '');
            }
            var div = document.querySelector('.product:nth-child('+num+') .product-around-image');
            div.dataset.price = object.price;
            var img = document.querySelector('.product:nth-child('+num+') .product-image');
            img.src = object.imageUrl;
            img.alt = object.text;
            var p = document.querySelector('.product:nth-child('+num+') .product-name');
            p.textContent = object.text;
            var output = document.querySelector('.product:nth-child('+num+') output[name*="total-price"]');
            output.textContent = object.price;
            var input = document.querySelector('.product:nth-child('+num+') input[name*="amount"]');
            input.id = 'a'+num;
            input.setAttribute('onclick', "recountPrice('input#a'+num+'")");
            input.setAttribute('onchange', "recountPrice('input#a'+num+'")");
            var button = document.querySelector('.product:nth-child('+num+') input[name*="to-center"]');
            button.id = 'b'+num;
            button.setAttribute('data-information', JSON.stringify(object));
            button.setAttribute('onclick', "setClassButton_sendToStorage('input#b'+num+'")");
        }
    }

    var clear = function() {

```

[illegible]

```
document.querySelector('.category:nth-child(10) .center:nth-child(2)').onclick = function()
{check_remove_add_sup('.category:nth-child(10) .center:nth-child(2)');getDataCreate();widthRegulate();catalogOnClick()}
document.querySelector('.category:nth-child(10) .center:nth-child(3)').onclick = function()
{check_remove_add_sup('.category:nth-child(10) .center:nth-child(3)');getDataCreate();widthRegulate();catalogOnClick()}
```

/*ДОВІДНИК КЛІЄНТІВ*/

```
var widthRegulate = function () {
    var klients = document.querySelector('.klients');
    var container = document.querySelector('.container');
    var category = document.querySelector('.category');
    var center = document.querySelector('.center');
    klients.style.width = (container.getBoundingClientRect().width - category.getBoundingClientRect().width -
center.getBoundingClientRect().width) + 'px';
}
```

/*ДОВІДНИК МАЙСТРІВ*/

```
var recountPrice = function (selector_input) {
    var input = document.querySelector(selector_input);
    var num = input.value;
    var output = input.nextElementSibling.nextElementSibling;
    var div = input.previousElementSibling.previousElementSibling;
    var price = div.dataset.price;
    output.textContent = Math.round(price * num * 100) / 100;
}
```

/*ДОВІДНИК ПОСЛУГ*/

```
var setClassButton_sendToStorage = function (selector_button) {
    var button = document.querySelector(selector_button);
    if (!button.parentElement.classList.contains("not-available") && !button.classList.contains("in-center")) {
        button.classList = "";
        button.classList.add("in-center");
        button.value = "";
        button.setAttribute('readonly', "");
        var input = button.previousElementSibling;
        input.setAttribute('disabled', "");
        input.setAttribute('title', "");
        addCenterValues(selector_button);
        sendToStorage(selector_button);
    }
}
```

/*ЗАПИС КЛІЄНТІВ*/

```
var addCenterValues = function (selector_button) {
    var center = document.querySelector('a.center');
    center.dataset.amount++;
    sessionStorage.setItem("I amount", center.dataset.amount);
    var output = document.querySelector(selector_button).nextElementSibling;
    var total = Math.round( (Number(center.dataset.total) + Number(output.value) ) * 100) / 100;
    center.dataset.total = total;
    sessionStorage.setItem("I total", center.dataset.total);
}
```

/*ПРАЙС-ЛИСТ ПОСЛУГ*/

```
var centerOnLoad = function () {
    var center = document.querySelector('a.center');
    if (sessionStorage.getItem("I amount")) {
        center.dataset.amount = sessionStorage.getItem("I amount");
    }
}
```

```

        center.dataset.total = sessionStorage.getItem("1total");
    }
    if (!sessionStorage.getItem("1amount")) {
        center.dataset.amount = 0;
    }
}

/*картка услуг*/

var catalogOnClick = function () {
    var name = document.querySelector('.active-super').dataset.supername;
    var keys = Object.keys(sessionStorage);
    keys.shift();
    keys.shift();
    var objects = keys.map(function(keys) {return JSON.parse(sessionStorage.getItem(keys));});
    for (var i = 0; i < objects.length; i++) {
        if (objects[i].centerName == name) {
            var button = document.querySelector(objects[i].buttonURL);
            button.classList.add("in-center");
            button.value = "";
            button.style.transitionProperty = "box-shadow";
            var amount = button.previousElementSibling;
            amount.value = JSON.parse(objects[i].amount);
            var selector_input = 'input#a' + objects[i].buttonURL[7];
            var amount = document.querySelector(selector_input);
            amount.setAttribute('disabled', "");
            amount.setAttribute('title', "");
            recountPrice(selector_input);
        }
    }
}

```

Файл main.html

```
<!DOCTYPE HTML>

<html lang="ukr">
<head>
    <meta charset="utf-8">
    <title>DeliveryService</title>
    <link href="/css/main_styles.css" rel="stylesheet">
</head>
<body onload="centerOnLoad()">

    <header class="header">
        <div class="container">
            <div class="flexbox-header">
                <div class="logo">
                    </img>
                </div>
                <ul class="list-menu">
                    <li>
                        <p>Контакти</p>
                        <ul class="numbers">
                            <li>+380671234567</li>
                                <li>+380501234567</li>
                                <li>+380931234567</li>
                            </ul>
                        </li>
                        <li><a href="order.html">НДІ</a></li>
                        <li><a href="partners.html">Вхідна інформація</a></li>
                        <li><a href="couriers.html">Вихідна інформація</a></li>
                    </ul>
                </div>
                <div>
                    <a href="center.html" class="center" data-amount="" data-total=""></a>
                </div>
            </div>
            <div class="move">
                <p>Вихід</p>
            </div>
        </div>
    </header>

    <footer class="footer">
        <div class="container">
            <div class="footer-middle">
                <div class="logo">
                    </img>
                </div>

                <div class="nav-list">
                    <h3>Контакти</h3>
                    <ul class="list-menu">
                        <li><p>+380671234567</p></li>
                            <li><p>+380501234567</p></li>
                            <li>
                                <p>+380931234567</p>
                            </li>
                        </ul>
                    </div>
                <div class="nav-list">
```

```
<h3>Наши</h3>
  <ul class="list-menu">
    <li><a href="partners.html">Партнеры</a></li>
    <li><a href="couriers.html">Курьеры</a></li>
  </ul>
</div>

  <div>

    <div>

      <a class="btn" href="order.html">Заказать</a>

    </div>

  </div>
<div class="footer-bottom">
  <div class="socials">
    <p>
      <b>Давайте дружить!</b>
      <a href="#">Фейсбук</a>
      <a href="#">Твиттер</a>
      <a href="#">Инстаграм</a>
    </p>
  </div>
  <div class="copyright">
    <p>@2025, Maria Riznyk</p>
  </div>
</div>
</div>
</footer>

<script src="./js/main_script.js" type="text/javascript"></script>
</body>
</html>
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ
Кафедра Комп'ютерних наук

ДИПЛОМНА РОБОТА
на ступінь вищої освіти бакалавр
із спеціальності 122 Комп'ютерні науки
Інформаційна система

Виконав: студентка 4 курсу, групи КНД-42
Проказа Богдан Ігорович
Керівник: доцент кафедри Комп'ютерних наук
к.т.н., доцент Прокопов С.В.

Київ - 2025

1

Слайд 1

АКТУАЛЬНІСТЬ РОБОТИ

2

Для ефективного розвитку сучасного підприємства необхідна здатність оперативно реагувати на зміни ринкового середовища. Більшість компаній вже перейшла до цифрових рішень, які дозволяють упорядковувати, структурувати та надійно зберігати великі масиви інформації. Сучасні клієнт-серверні технології спрощують процес обробки та ведення даних, оскільки ґрунтуються на централізованому зберіганні й комплексному аналізі інформаційних масивів. Розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі дозволить ефективно збирати оперативну інформацію для обліку та подальшого аналізу. Тому тема розробленої бакалаврської роботи являється *актуальною*.

Слайд 2

МЕТА, ОБ'ЄКТ ТА ПРЕДМЕТ ДОСЛІДЖЕННЯ

3

Об'єкт дослідження – спортивно-оздоровчий комплекс.

Предмет дослідження – інформаційна система для контролю абонементів в спортивно-оздоровчому комплексі.

Мета роботи – розробити інформаційну систему для обліку та контролю абонементів в спортивно-оздоровчому комплексі, призначену для надання оперативної інформації для обліку і подальшого її аналізу та зниження обсягів паперового документообігу.

Наукове завдання – аналіз інформаційних систем, їх класифікація, архітектура та створення; дослідження інформаційних процесів при контролі абонементів спортивно-оздоровчого комплексу; розробка інформаційної системи для контролю абонементів спортивно-оздоровчого комплексу.

Методи дослідження – HTML, CSS, JavaScript для клієнтської частини, Node.js, AJAX, Firebird Database для серверної частини.

Слайд 3

1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

4

Інформаційна система - комплекс взаємопов'язаних економічних задач, що характеризуються інтенсивним обміном інформацією між складовими елементами.

З метою реалізації основних управлінських функцій виділяють *чотири ключові підсистеми*:

- планування – формування стратегій і тактичних рішень;
- оперативне управління – координація поточної діяльності;
- облік – фіксація та систематизація даних про господарські операції;
- аналіз – оцінка ефективності діяльності й прийняття управлінських рішень.

Слайд 4

1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 5

Інформаційні системи можна **класифікувати** за масштабами й функціональними можливостями на три основні класи:

- малі інформаційні системи;
- середні інформаційні системи;
- великі (корпоративні) інформаційні системи, що обслуговують потреби організацій регіонального або національного рівня.

Інформаційну систему як комплекс взаємопов'язаних складових можна **класифікувати** за такими рівнями:

- апаратний рівень: включає комп'ютери, периферійне устаткування, мережеву та телекомунікаційну інфраструктуру;
- системний та системно-залежний рівень: охоплює операційні системи, мережеві протоколи тощо;
- рівень прикладного середовища: охоплює middleware-рішення (зокрема CORBA, DCE, Tuxedo), системи керування базами даних (DBMS), Intranet-технології, OLAP-сервіси та комунікаційні інтерфейси.

Слайд 5

1 ОГЛЯД ТА АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ 6

Архітектура інформаційних систем

Дворівнева архітектура передбачає наявність двох рівнів взаємодії. Один з компонентів виконує роль сервера, що надає певний набір сервісів, якими, у свою чергу, користується клієнтський компонент. Останній ініціює звернення до сервера з метою отримання необхідних послуг у ході виконання прикладних задач.

Трирівнева архітектура вирізняється меншою жорсткістю зв'язків між клієнтськими та серверними частинами та більш гнучко обробляє дані. Модель має три логічно незалежні компоненти:

- рівень представлення (інтерфейс користувача),
- прикладний рівень (сервер додатків),
- рівень доступу до інформаційних ресурсів (база даних).

Розподілена однорангова архітектура: клієнт вже не обмежується лише функцією представлення або інтерфейсом користувача. Він може реалізовувати прикладну логіку, а в окремих випадках – навіть виступати як сервер додатку. Як клієнт, так і сервер набувають властивостей взаємозамінності та рівноправності, формуючи однорангове середовище взаємодії.

Слайда 6

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ 7

Firebird – реляційна система керування базами даних, потужна SQL-сумісна система для зберігання, обробки та аналізу структурованих даних. Firebird працює на багатьох операційних системах – Windows, Linux, macOS, FreeBSD – і підтримує потужні механізми транзакцій, запитів, індексації та безпеки.

Основні характеристики:

- Реляційна модель: Дані організовані у вигляді таблиць, пов'язаних між собою через ключі.
- Мова SQL: Firebird підтримує стандартну мову запитів SQL, включаючи транзакції, процедури, тригери та представлення.
- Firebird не потребує постійного адміністрування – це робить її чудовим варіантом для вбудованих систем або невеликих компаній.
- Вбудована або серверна архітектура: Працює як локально, так і у вигляді сервера.

Слайд 7

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ 8

- HTML5 визначає структуру сторінки. Визначає розташування тексту, зображень, відео, посилань і інших елементів на сторінці за допомогою тегів, які надають контенту специфічну семантику.
- CSS3 відповідає за стильове оформлення сторінки. Для досягнення стилізації веб-сторінок CSS надає ряд зручних можливостей: вибір кольорів, шрифтів, розмірів елементів, визначення відступів та границь, використання селекторів.
- JavaScript надає сайту інтерактивності й динаміки. Можливості Javascript: обробка подій, маніпуляція dom, валидація форм, анімація й візуальні ефекти, взаємодія із сервером.

Слайд 8

2 ІНСТРУМЕНТАЛЬНІ ЗАСОБИ ДЛЯ РОЗРОБКИ ІНФОРМАЦІЙНОЇ СИСТЕМИ ⁹

Інтеграція клієнтської та серверної частин.

Сучасні веб-застосунки складаються з двох основних частин:

- *Клієнтська частина* (frontend) – те, що бачить і з чим взаємодіє користувач (HTML, CSS, JavaScript);
- *Серверна частина* (backend) – логіка, яка працює "за лаштунками": обробка запитів, доступ до баз даних, авторизація тощо.

Щоб ці частини "спілкувалися", використовують технології передачі даних, зокрема AJAX, технологія, яка дозволяє надсилати запити на сервер без перезавантаження сторінки. У ролі сервера часто виступає Node.js – сучасне середовище виконання JavaScript, яке дозволяє створювати високопродуктивні сервери на тій же мові, що й фронтенд.

Слайд 9

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ ¹⁰

Меню інформаційної системи включає форми нормативно-довідкової інформації (НДІ), а також вхідні та вихідні екранні форми.

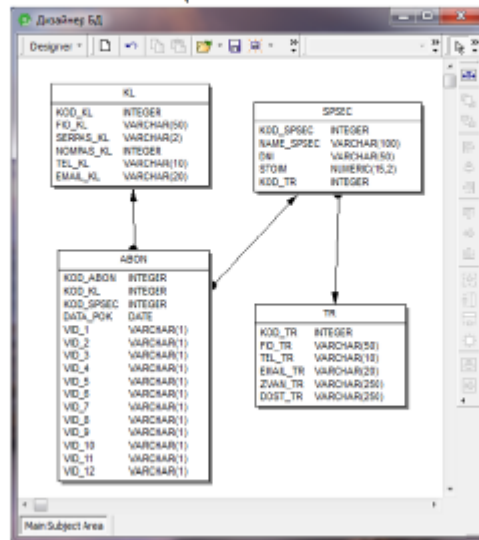


Структура меню інформаційної системи

Слайд 10

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

11



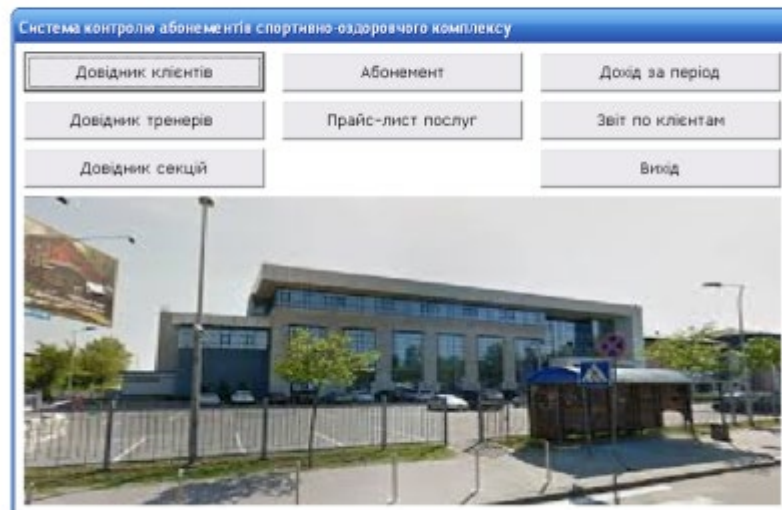
Діаграма бази даних

11

Слайд 11

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

12



Головна форма інформаційної системи

12

Слайда 12

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ Форми нормативно-довідкової інформації (НДІ)

13

Довідник клієнтів

ПІБ	Серія паспорту	Номер паспорту	Телефон	E-mail
Дітко Андрій Федорович	КУ	896785	(068)7766762	slip@gmail.com
Мартинський Анатолій Володимирович	МЦ	129785	(068)5433399	mat@gmail.com
Дітков Дмитро Юрійович	АЕ	679545	(068)2100000	slip@gmail.com
Солошенко Микола Павлович	ОЛ	879554	(068)2220760	solosh@gmail.com
Сенюченко Юрій Павлович	ПР	789554	(068)2100000	senyouch@gmail.com
Сусак Ярослав Михайлович	ПН	453245	(068)2100000	suhak@gmail.com
Селіда Сергій Федорович	ПР	567433	(068)2155505	selida@gmail.com
Селіда Катерина Григорівна	НБ	567432	(068)2155505	selida@gmail.com
Богданов Віталій Олександрович	АЕ	897678	(068)4451718	bag@gmail.com
Понітовський Олександр Сергійович	КУ	768978	(068)5100081	ponosov@gmail.com
Бондаров Володимир Володимирович	УБ	768567	(068)5100775	bondar@gmail.com
Боліщій Євген Євгенович	НП	675456	(068)5100000	bolish@gmail.com

Пошук клієнта



Довідник клієнтів

13

Слайд 13

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ Форми нормативно-довідкової інформації (НДІ)

14

Довідник клієнтів

ПІБ	Серія паспорту	Номер паспорту	Телефон	E-mail
Булбаский Віктор Володимирович	АП	342342	(068)6655551	bulb@gmail.com
Либонич Ілля Євгенович	АЕ	828615	(068)2571112	libon@gmail.com
Богданов Віталій Олександрович	АЕ	897678	(068)4451718	bag@gmail.com
Бондаров Володимир Володимирович	УБ	768567	(068)5100775	bondar@gmail.com
Боліщій Євген Євгенович	НП	675456	(068)5100000	bolish@gmail.com

Пошук клієнта



Пошук клієнта в Довіднику клієнта

14

Слайд 14

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

15

Форми нормативно-довідкової інформації (НДІ)

Довідник тренерів

ПІБ	Телефон	E-mail
Костенко Михайло Петрович	095 0660099	kostenko.mikhail@gmail.com
Пирітень Валерія Михайлівна	097 7770900	
Король Михайло Сергійович	097 0970979	
Федорук Наталія Євгенівна	098 0990000	natja_nf@ukr.com
Бурко Сергій Валерійович	099 1011191	ser_burko@gmail.com
Муратов Валерій Артемійович	093 4600000	
Гладкова Світлана Валентинівна	098 0700444	

Звання: майстер спорту України міжнародного класу з футболу

Досвідчення: членом секції серед студентів з міні-футболу в складі збірної команди України

Вихід



Довідник тренерів

15

Слайд 15

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

16

Форми нормативно-довідкової інформації (НДІ)

Довідник секцій

Назва секції	Дні відвідувань	Вартість занять на місяць, грн.
футбол (юношки)	Вт, Чт, Сб	300
міні-футбол (жінки)	Вт, Чт, Сб	300
міні-футбол (чоловіки)	Вт, Чт, Сб	300
бейсбол	Пн, Ср, Пт	250
аerобічна гімнастика (бодібілдінг)	Вт, Чт, Сб	570
рукопашний бй	Пн, Ср, Пт	330
сенбо і сойбоє сенбо	Вт, Чт, Сб	560
фітнес	Вт, Чт, Сб	750
боротьба спортивна	Пн, Ср, Пт	890
танцювальна аеробіка	Вт, Чт, Сб	300

Тренер: Гордієва Світлана Валентинівна

Обрати тренера

Вихід



Довідник
спортивних секцій

16

Слайд 16

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ Вхідна інформація

17

Меню																	
Оформити новий абонемент			Пошук клієнта:			Пошук роздрувати абонемент			Вихід								
№	ІМ'Я КЛІЄНТА	ІМЕНА	Вартість заняття на місяць, грн.	Діючий до ..	Дні відвідувань	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
22	Васильєв Александр Андреевич	футбол (чоловік)	600	31.04.2025	Пн, Ср, Пт	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐
23	Галущак Сергій Миколайович	спортивна аеробика	600	09.04.2025	Вт, Чт, Сб	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐
24	Шеко Андрій Федорович	теніс настільний	600	01.05.2025	Пн, Ср, Пт	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐
25	Галущак Сергій Миколайович	теніс настільний	600	01.05.2025	Пн, Ср, Пт	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐
26	Балашовий Євген Віталійович	спортивне орієнтування	300	02.05.2025	Пн, Ср, Пт	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐
27	Маровський Дієн Володимирович	теніс футбол (чоловік)	600	30.05.2025	Вт, Чт, Сб	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☐	☐

Абонемент

17

Слайд 17

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ Вхідна інформація

18

Оформити новий абонемент

Обрати клієнта

Обрати секцію

Дні відвідувань

Вартість заняття на місяць, грн.

Зберегти

Відмінити

Оформити новий абонемент

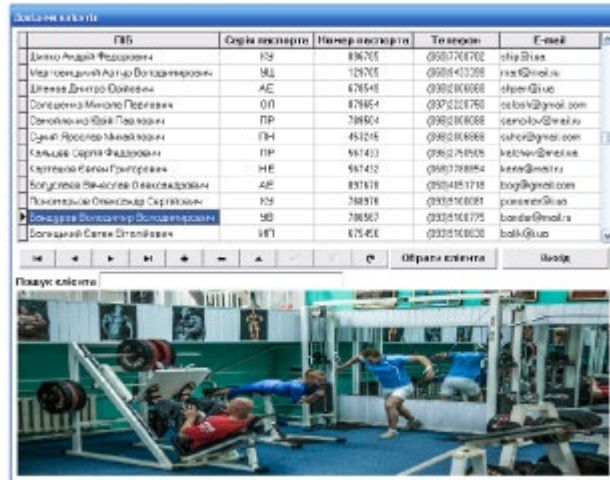
18

Слайда 18

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

19

Вхідна інформація



Обрати клієнта

19

Слайд 19

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

20

Вхідна інформація



Обрати секцію

20

Слайд 20

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

21

Вхідна інформація

Зберегти новий абонемент

21

Слайд 21

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

22

Вхідна інформація

Новий абонемент

22

Слайд 22

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

23

Вхідна інформація

Меню n1																	
Оформити новий абонемент			Пошук клієнта:		Повернуто роздрукувати абонемент										Вихід		
№	ПІБ клієнта	Назва клубу	Вартість запису на місяць, грн.	Дієвий до	Дні вихідні	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
25	Галушко Сергій Михайлович	теніс настільний	600	01.05.2025	Пн, Ср, Пт	☐	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐
26	Бондаренко Олег Валентинович	спортивне орієнтування	300	02.05.2025	Пн, Ср, Пт	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
27	Морозовський Дмитро Володимирович	теніс-футбол (застава)	600	05.05.2025	Вт, Чт, Сб	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
28	Радвак Ірина Іванівна	боротьба спортивна	600	06.05.2025	Пн, Ср, Пт	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
29	Морозовський Дмитро Володимирович	парашей спорт	600	09.07.2025	Вт, Чт, Сб	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
31	Бондаренко Володимир Володимирович	боротьба спортивна	600	09.07.2025	Пн, Ср, Пт	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Доданий абонемент

Меню n1																	
Оформити новий абонемент			Пошук клієнта: 31			Повернуто роздрукувати абонемент										Вихід	
№	ПІБ клієнта	Назва клубу	Вартість запису на місяць, грн.	Дієвий до	Дні вихідні	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
25	Бондаренко Сергій Михайлович	спортивна боротьба	600	01.05.2025	Пн, Ср, Пт	☒	☒	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐
31	Бондаренко Володимир Володимирович	боротьба спортивна	600	09.07.2025	Пн, Ср, Пт	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐
[4]																	

Пошук клієнта

23

Слайд 23

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

24

Вхідна інформація

Меню n1																	
Оформити новий абонемент			Пошук клієнта: 31			Повернуто роздрукувати абонемент			Вихід								
№	ПІБ клієнта	Назва клубу	Вартість запису на місяць, грн.	Дієвий до	Дні вихідні	Відвідування											
						1	2	3	4	5	6	7	8	9	10	11	12
31	Бондаренко Володимир Володимирович	боротьба спортивна	600	09.07.2025	Пн, Ср, Пт	☒	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐	☐

Відвідування клієнта

24

Слайда 24

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

25

Вхідна інформація

Абонемент

Абонемент №

Где куплен

Номер ордера

Возраст абонента на 1 января, лет

Дейский до ...

Видеоуслуга

☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6

☐ 7 ☐ 8 ☐ 9 ☐ 10 ☐ 11 ☐ 12



Повторно роздрукувати
абонемент при наявності змін

25

Слайд 25

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

26

Вхідна інформація

Прайс-лист услуг		
Наименование	Договорная	Фиксированная
Анализ рынка	Р/д, С/д, П/д	500
Анализ конкурентов	Р/д, М/д, С/д	400
Анализ финансовых показателей	Р/д, С/д, П/д	600
Анализ маркетинга	Р/д, М/д, С/д	700
Анализ сбыта	Р/д, М/д, С/д	300
Анализ логистики	Р/д, С/д, П/д	300
Анализ спроса	Р/д, М/д, С/д	600
Анализ качества	Р/д, С/д, П/д	700
Анализ рисков	Р/д, М/д, С/д	500
Анализ эффективности	Р/д, С/д, П/д	300
Анализ маркетинга	Р/д, М/д, С/д	600
Анализ продаж	Р/д, С/д, П/д	600
Анализ качества продукции	Р/д, М/д, С/д	600
Анализ	Р/д, С/д, П/д	700
Анализ качества продукции	Р/д, М/д, С/д	500
Анализ качества	Р/д, С/д, П/д	600
Анализ качества	Р/д, М/д, С/д	600
Анализ качества	Р/д, С/д, П/д	600
Анализ качества	Р/д, М/д, С/д	600

Прайс-лист услуг

28

Слайд 26

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

27

Вихідна інформація



Дохід за період

27

Слайд 27

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

28

Вихідна інформація

Доход за період з 09.03.2025 по 09.06.2025				
№	ІМ'Я КЛІЄНТА	НАВІВ СЕКТОР	ВІСІ	Вартість надання послуги
24	Шенко Андрій Федорович	теніс настільний	800	01.04.2025
25	Головченко Сергій Миколайович	теніс настільний	800	01.04.2025
26	Болдирєв Віктор Вікторович	спортивне орієнтування	300	01.04.2025
27	Маршак Андрій Володимирович	нині-футбол (чоловіки)	800	31.03.2025
28	Головченко Сергій Миколайович	боротьба спортивна	800	29.03.2025

Сформувати звіт
про дохід за період

Друк звіту
про дохід за період

Доход за період з 09.03.2025 по 09.06.2025				
№	ІМ'Я КЛІЄНТА	НАВІВ СЕКТОР	ВІСІ	Вартість надання послуги
24	Шенко Андрій Федорович	теніс настільний	800	01.04.2025
25	Головченко Сергій Миколайович	теніс настільний	800	01.04.2025
26	Болдирєв Віктор Вікторович	спортивне орієнтування	300	01.04.2025
27	Маршак Андрій Володимирович	нині-футбол (чоловіки)	800	31.03.2025
28	Головченко Сергій Миколайович	боротьба спортивна	800	29.03.2025

Слайд 28

3 ОПИС ПРОГРАМНОГО ПРОДУКТУ - ІНФОРМАЦІЙНОЇ СИСТЕМИ

29

Вихідна інформація

Звіт по клієнтам				
№	П.Б.І. клієнта	Назва секції	Вартість	на-Дата розрахунку
22	Басіненко Анатолій Андрійович	рукопашний бій	600	31.02.2025
23	Жуковський Сергій Миколайович	Окреслення 1800м	600	31.02.2025
24	Шаталов Олександр	теніс настільний	600	31.04.2025
25	Галущак Сергій Миколайович	теніс настільний	600	31.04.2025
26	Семидинов Олег Вікторович	спортивна орієнтування	300	31.04.2025
27	Мартинюк Артур Володимирович	найт-футбол (чоловіки)	600	31.03.2025
28	Павлов Васильович Кривошея	боротьба спортивна	600	31.04.2025
29	Мартинюк Артур Володимирович	рукопашний спорт	600	31.04.2025
30	Бондаренко Володимир Володимирович	боротьба спортивна	600	31.04.2025

Звіт по клієнтам

Друк звіту по клієнтам

Звіт по клієнтам				
№	П.Б.І. клієнта	Назва секції	Вартість	на-Дата розрахунку
22	Басіненко Анатолій Андрійович	рукопашний бій	600	31.02.2025
23	Жуковський Сергій Миколайович	Окреслення 1800м	600	31.02.2025
24	Шаталов Олександр	теніс настільний	600	31.04.2025
25	Галущак Сергій Миколайович	теніс настільний	600	31.04.2025
26	Семидинов Олег Вікторович	спортивна орієнтування	300	31.04.2025
27	Мартинюк Артур Володимирович	найт-футбол (чоловіки)	600	31.03.2025
28	Павлов Васильович Кривошея	боротьба спортивна	600	31.04.2025
29	Мартинюк Артур Володимирович	рукопашний спорт	600	31.04.2025
30	Бондаренко Володимир Володимирович	боротьба спортивна	600	31.04.2025

Слайд 29

ВИСНОВКИ

30

Метою роботи була розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі, призначеної для надання оперативної інформації для обліку і подальшого її аналізу та зниження обсягів паперового документообігу.

Для її реалізації було зроблено:

Зроблено аналіз інформаційних систем, вивчені їх класифікація, архітектура та створення.

Досліджено інформаційні процеси при контролі абонементів спортивно-оздоровчого комплексу.

За допомогою технологій HTML, CSS, JavaScript для створення клієнтської частини та технологій Node.js, AJAX, Firebird Database для створення серверної частини розроблена інформаційна система для контролю абонементів спортивно-оздоровчого комплексу, призначена для надання оперативної інформації для обліку та подальшого аналізу.

Слайда 30

ВИСНОВКИ

31

Розробка інформаційної системи для обліку та контролю абонементів в спортивно-оздоровчому комплексі включає:

- створення бази даних для інформаційної системи;
- створення форм для надання нормативно-довідникової інформації (довідники клієнтів, тренерів та спортивних секцій);
- створення вхідних документів (створення абоненту з можливістю його друку, прайс-лист послуг);
- створення вихідних документів (дохід за період, звіт по клієнтам з можливістю друку цих документів).

Розробка інформаційної системи буде сприяти підвищенню продуктивності праці співробітників спортивного комплексу, оскільки надасть можливість оперативно збирати та аналізувати інформацію.

Слайд 31