

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ ЕКОНОМІКИ ТА БІЗНЕС АДМІНІСТРУВАННЯ
КАФЕДРА БІЗНЕС АНАЛІТИКИ ТА ЦИФРОВОЇ ЕКОНОМІКИ

ДОПУСТИТИ ДО ЗАХИСТУ
Завідувач кафедри бізнес-аналітики
та цифрової економіки
_____ Наталія КАСЬЯНОВА
« ___ » _____ 2023р.

КВАЛІФІКАЦІЙНА РОБОТА
(ПОЯСНЮВАЛЬНА ЗАПИСКА)
ЗДОБУВАЧА ОСВІТНЬОГО СТУПЕНЯ «МАГІСТР»
ЗІ СПЕЦІАЛЬНОСТІ 051 «ЕКОНОМІКА»
ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «ЕКОНОМІЧНА КІБЕРНЕТИКА»

Тема: «Оптимізація управління базами даних в бізнесі»

Виконав Артем СЕВЕРИНЕНКО

Керівник: д.е.н., професор Наталія КАСЬЯНОВА

Консультанти з розділів:

Розділ 1: д.е.н., професор Наталія КАСЬЯНОВА

Розділ 2: д.е.н., професор Наталія КАСЬЯНОВА

Розділ 3: д.е.н., професор Наталія КАСЬЯНОВА

Нормоконтролер із ЄСКД (ЄСПД):

ст. викладач Юлія Дияк

Київ 2023

НАЦІОНАЛЬНИЙ АВІАЦІЙНИЙ УНІВЕРСИТЕТ

Факультет економіки та бізнес-адміністрування

Кафедра бізнес-аналітики та цифрової економіки

Спеціальність: 051 «Економіка»

Освітньо-професійна програма «Економічна кібернетика»

ЗАТВЕРДЖУЮ

Завідувач кафедри бізнес-аналітики та цифрової економіки

_____ Наталія Касьянова

« ____ » _____ 2023 р.

ЗАВДАННЯ НА КВАЛІФІКАЦІЙНУ РОБОТУ

Здобувача вищої освіти: СЕВЕРИНЕНКА Артема

Тема роботи: «Оптимізація управління базами даних в бізнесі»
затверджена наказом ректора № 1085/ст від 07.07.2023р.

1. Термін здачі студентом закінченої роботи на кафедру: 22.12.2023 р.
2. Вихідні дані до роботи: бази даних, бізнес-моделі, електронні інформаційні джерела
3. Зміст дослідження: аналітика використання баз даних в товарному бізнесі, формулювання схем праці співробітників з базами даних, формулювання вимог співробітників до системи. Розробка власної підсистеми CRM, що відповідатиме за роботу з базами даних та забезпечуватиме безпеку даних.
4. Перелік обов'язкових демонстраційних матеріалів: ... слайдів

5. Календарний план-графік

№ пор.	Назва етапів випускної роботи	Термін виконання	Позначки про виконання
1.	Отримання завдання на випускну роботу	22.09.2023	<i>виконано</i>
2.	Огляд літератури за темою	25.09.2023	<i>виконано</i>
3.	Оцінка негативних та позитивних сторін використання наявних систем.	30.09.2023	<i>виконано</i>
4.	Обґрунтування необхідності введення змін в систему (використання продукту).	4.10.2023	<i>виконано</i>
5.	Пропрацювання деталей продукту.	15.10.2023	<i>виконано</i>
6.	Оцінка вартості продукту.	18.10.2023	<i>виконано</i>
7.	Підрахунок вартості введення змін в систему для різних моделей бізнесу з різними підходами	22.10.2023	<i>виконано</i>
8.	Проведення порівняльної характеристики з боку витрат, часу	29.10.2023	<i>виконано</i>
9.	Аналіз отриманих результатів	05.12.2023	<i>виконано</i>
10.	Розробка слайдів та написання доповіді	09.12.2023	<i>виконано</i>
11.	Попередній захист випускної роботи	10.12.2023	<i>виконано</i>
12.	Корегування роботи за результатами попереднього захисту	11.12.2023	<i>виконано</i>
13.	Остаточне оформлення випускної роботи та слайдів	13.12.2023	<i>виконано</i>
14.	Підписання відгуку та рецензії	14.12.2023	<i>виконано</i>
15.	Захист випускної роботи у ДЕК	25.12.2023	<i>виконано</i>

Дата видачі завдання: 22.09.2023р.

Керівник випускної роботи _____ Наталія КАСЬЯНОВА

Завдання прийняв до виконання _____ Артем СЕВЕРИНЕНКО

РЕФЕРАТ

СЕВЕРИНЕНКО Артем. Оптимізація управління базами даних в бізнесі. - Випускна робота зі спеціальності 051 «Економіка», ОПП «Економічна кібернетика». Національний авіаційний університет Міністерства освіти і науки України, м. Київ, 2023.

Випускна робота містить 100 сторінок, 8 таблиць, 14 рисунків, список використаних джерел з 47 найменувань.

Об'єкт дослідження – процеси використання баз даних в товарному бізнесі, їх захисту та зручності використання.

Предметом дослідження є бази даних, що використовуються для щоденної роботи працівників різного рангу в компаніях, що працюють в сфері товарного бізнесу.

Мета дослідження – розробка пропозиції і варіанту реалізації впровадження нової структури роботи системи управління бази даних, яка виділятиметься підвищеною безпекою даних та зручністю користування

При написанні роботи використовувалися методи дослідження: аналіз та синтез, порівняльні та статистичні методи, розробка програмного забезпечення

Ключові слова: безпека баз даних, зручність використання баз даних, СКБД, CRM, бази даних в товарному бізнесі, СКД, ББД

ABSTRACT

SEVERYNENKO Artem. Optimization of Database Management in Business. - Graduation thesis in the field of 051 "Economics," Educational Program "Economic Cybernetics." National Aviation University of the Ministry of Education and Science of Ukraine, Kyiv, 2023.

The graduation thesis comprises 90 pages, 12 tables, 14 figures, a list of used sources with 59 references.

The object of the research is the processes of using databases in the commodity business, their security, and usability.

The subject of the study is databases used for the daily work of employees of various ranks in companies operating in the field of commodity business.

The research goal is to develop a proposal and implementation variant for introducing a new structure of the database management system, distinguished by enhanced data security and user convenience.

Research methods employed in the work include analysis and synthesis, comparative and statistical methods, and software development.

Keywords: database security, usability of databases, DBMS, CRM, databases in commodity business, DBS, RDBMS.

ЗМІСТ

ВСТУП	8
РОЗДІЛ 1 ТЕОРЕТИЧНІ ОСНОВИ ФОРМУВАННЯ БАЗ ДАНИХ В БІЗНЕСІ	10
1.1 Сутність та структура баз даних	10
1.2 Особливості управління базами даних	22
1.3 Технологія використання СУБД	26
Висновки до першого розділу	33
РОЗДІЛ 2 МЕТОДИЧНІ ПІДХОДИ ДО ОПТИМІЗАЦІЇ БАЗ ДАНИХ	35
2.1. Методи масштабування реляційних баз даних	35
2.2. Методичні підходи до оптимізації баз даних	44
2.3. Кібербезпека даних як складова оптимізації реляційних баз даних	49
Висновки до другого розділу	61
РОЗДІЛ 3 ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ОПТИМІЗАЦІЇ БАЗИ ДАНИХ В БІЗНЕСІ	63
3.1. Створення автоматизованої інформаційної системи підприємства	63
3.2 Проектування програмного забезпечення системи обліку діяльності фітнес-клубу	76
3.3. Планування розробки та оцінка бюджету	89
Висновки до третього розділу	95
ВИСНОВКИ	96
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	98

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

ПЗ – програмне забезпечення;

ПК – персональний комп'ютер;

ОС – операційна система;

БД – база даних;

ТБ - товарний бізнес;

СКБД - система управління базами даних

СКД - система керування даними

ББД - безпека баз даних

УІБ - система управління інформаційною безпекою

КЗСІ - комплексна система захисту інформації

DLP - запобігання втраті даних

DNS - система доменних імен

ІВ - інтелектуальна власність

ВСТУП

Бази даних в сучасному бізнесі відіграють важливу роль, надаючи компаніям можливість зберігання та обробки величезних обсягів даних. Це не просто технологія, але ключовий інструмент для прийняття виважених рішень та підтримки стратегічного розвитку. У цій статті розглянемо актуальний стан використання баз даних в бізнесі та їх вплив на ефективність і конкурентоспроможність компаній.

Бази даних дозволяють компаніям зберігати та керувати різноманітною інформацією, включаючи дані про клієнтів, фінансові та виробничі процеси. Це створює можливість ефективного управління даними та швидкого доступу до них. Бази даних допомагають у визначенні ключових трендів, аналізі конкурентів та розробці стратегій розвитку. Вони стають основою для прийняття стратегічних рішень, що сприяє успішному розвитку бізнесу.

Однією з сучасних тенденцій є використання хмарних баз даних. Це дозволяє компаніям зберігати та обробляти дані онлайн, забезпечуючи доступність та масштабованість.

Розширення обсягів даних викликає потребу в аналізі великих обсягів інформації. Бази даних, що підтримують великі дані, дозволяють використовувати цю інформацію для прогнозування та прийняття стратегічних рішень.

У сфері роздрібної торгівлі бази даних використовуються для ведення клієнтської бази та аналізу купівельних звичок, що допомагає виробникам пристосовувати свою пропозицію до потреб споживачів. У виробництві бази даних використовуються для ведення обліку сировини, контролю якості та планування виробничих процесів. В логістиці вони допомагають відстежувати рух товарів та оптимізувати постачання.

Зростання кількості кіберзагроз вимагає від компаній вдосконалення систем захисту баз даних. Захищеність інформації стає критично важливою для уникнення витоків даних та інших кібератак. Інтеграція баз даних з

інструментами штучного інтелекту та машинного навчання відкриває нові перспективи для аналізу даних та автоматизації процесів прийняття рішень.

Підсумовуючи, бази даних в сучасному бізнесі стали невід'ємною частиною ефективного управління та стратегічного розвитку. Облачні технології, аналітика великих даних, інтеграція з інтелектуальними системами - це лише деякі напрямки розвитку, які визначають актуальний стан використання баз даних в бізнесі. З урахуванням викликів та перспектив, компанії, які правильно використовують ці технології, зможуть підтримувати свою конкурентоспроможність та розвиватися в умовах постійних змін на ринку.

Мета дослідження – розробка пропозиції і варіанту реалізації впровадження нової структури роботи системи управління бази даних, яка виділятиметься підвищеною безпекою даних та зручністю користування

Для досягнення мети в роботі вирішено наступні задачі:

1. Проведена аналітика сучасних CRM-систем та СКБД
2. Визначені слабкі місця безпеки сучасних CRM-систем та СКБД
3. Розглянуті сучасні рішення, пов'язані з безпекою даних
4. Оформлені шляхи їх інтеграції в СКБД та CRM.

Об'єкт дослідження – слабкі місця баз даних бізнесу, їх захисту та зручності використання.

Предметом дослідження є бази даних, що використовуються для щоденної роботи працівників різного рангу в компаніях, що працюють в сфері товарного бізнесу.

При написанні роботи використовувалися методи дослідження: аналіз та синтез, порівняльні та статистичні методи

РОЗДІЛ 1

ТЕОРЕТИЧНІ ОСНОВИ ФОРМУВАННЯ БАЗ ДАНИХ В БІЗНЕСІ

1.1 Сутність та структура баз даних

У світлі стрімкого технологічного розвитку та зростання об'єму цифрової інформації, найбільш гострою проблемою для сучасних систем керування базами даних (СКБД) стає питання безпеки і захисту конфіденційності даних. Це питання набуває особливого значення, оскільки сучасні організації високо цінують свою інформацію як найцінніший актив, а втрата або пошкодження цієї інформації може мати подолання погіршуючі наслідки.

Зростання кількості кіберзагроз, таких як хакерські атаки, витоки даних та зловживання інформацією, підкреслює важливість вдосконалення безпекових механізмів у СКБД. Не тільки масштабні корпорації, а й менші підприємства та стартапи стають об'єктом цифрових загроз, що підкреслює необхідність впровадження комплексних стратегій безпеки.

У зусиллях забезпечити безпеку даних, використання сучасних методів шифрування стає вирішальним аспектом. Захист від несанкціонованого доступу до бази даних через шифрування конфіденційної інформації на рівні файлів і передачах дозволяє уникнути витоків даних під час їх трансляції через мережу.

Ефективна система контролю доступу, що включає в себе різні рівні привілеїв для користувачів, забезпечує обмеження доступу лише до необхідної інформації, зменшуючи ризик зловживання правами доступу та несанкціонованого витоку даних.

Регулярний моніторинг та аудит безпеки, включаючи ведення детального журналу подій, дозволяють оперативно виявляти та реагувати на потенційні загрози безпеки. Додатково, здатність встановлювати складні паролі та їх хешування сприяє створенню міцного бар'єру проти несанкціонованого доступу.

Навіть найбільш ефективна технічна інфраструктура може бути компрометована, якщо вона не супроводжується відповідною культурою безпеки серед персоналу. Тому важливо залучати персонал до процесів забезпечення безпеки, навчати його впізнавати потенційні загрози та дотримуватися найкращих практик в області кібербезпеки.

Усі ці заходи в сукупності допомагають створити необхідні умови для оптимальної та якісної роботи з базою даних, забезпечуючи найвищий рівень безпеки та захисту для інформації компанії.

Розберемось детальніше з тим, що таке база даних, система управління базами даних тощо.

База даних – це набір даних, організований відповідно до концепції, що описує характеристики цих даних і взаємозв'язок між їх елементами. Загалом база даних містить схеми, таблиці, графічні уявлення, збережені процедури та інші об'єкти. Дані в базі даних організовані відповідно до моделі організації даних. Таким чином, сучасна база даних, крім самих даних, містить їх опис і може містити засоби їх обробки.

Бази даних використовуються повсюдно в сучасному світі – в державних установах, у бізнесі, при структуруванні персональних даних і т. д. Саме тому бази даних максимально різні. А для більш чіткого розуміння різновидів баз даних – їх розрізняють за типами. Існує 3 основних типи баз даних:

1. Прості структури даних

Першим і найпростішим способом зберігання даних є текстові файли. Метод сьогодні використовується для роботи з невеликими обсягами інформації.

2. Ієрархічні бази даних

На відміну від текстових таблиць, у наступному типі БД є зв'язки між об'єктами. В ієрархічних базах даних кожен запис має одного «батька». Це створює деревовидну структуру, в якій записи класифікуються за їх відношенням до ланцюжка батьківських записів

Наслідки:

- інформація організована у вигляді дерева стосунків між батьками та дітьми;
- кожен запис може мати не більше одного з батьків;
- зв'язки між записами здійснюються за допомогою фізичних покажчиків;
- неможливо реалізувати відношення багато до багатьох.

Приклади:

- файлові системи
- система доменних імен (DNS)
- мережевий протокол прикладного рівня (LDAP)

3. Мережні бази даних розширюють функціональні можливості ієрархічних баз даних: записи можуть мати більше одного батьківського (рис. 1.1). Це означає, що складні відносини можна моделювати.

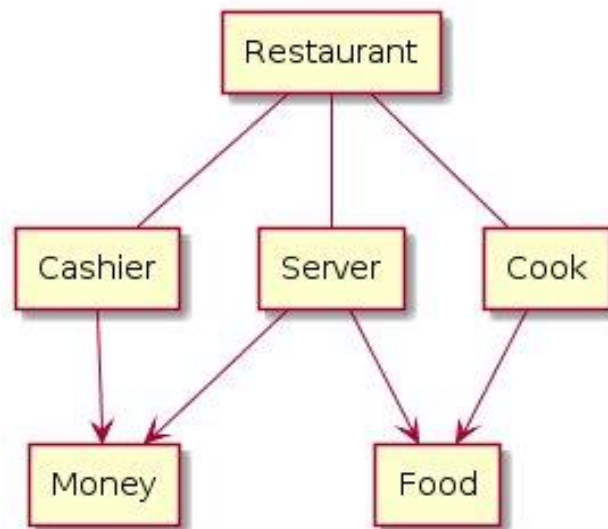


Рис. 1.1. Приклад побудови моделі мережевої бази даних

Наслідки:

- Мережеві бази даних подаються не деревом, а загальним графом
- обмежені тими ж шаблонами доступу, що ієрархічні БД
- Система керування даними (IDMS)

4. SQL бази даних

Реляційні бази даних – найстаріший тип досі широко використовуваних БД загального призначення (рис. 1.2). Дані та зв'язки між даними організовані за допомогою таблиць. Кожен стовпець у таблиці має ім'я і тип. Кожен рядок представляє окремий запис або елемент даних в таблиці, який містить значення для кожного з стовпців.

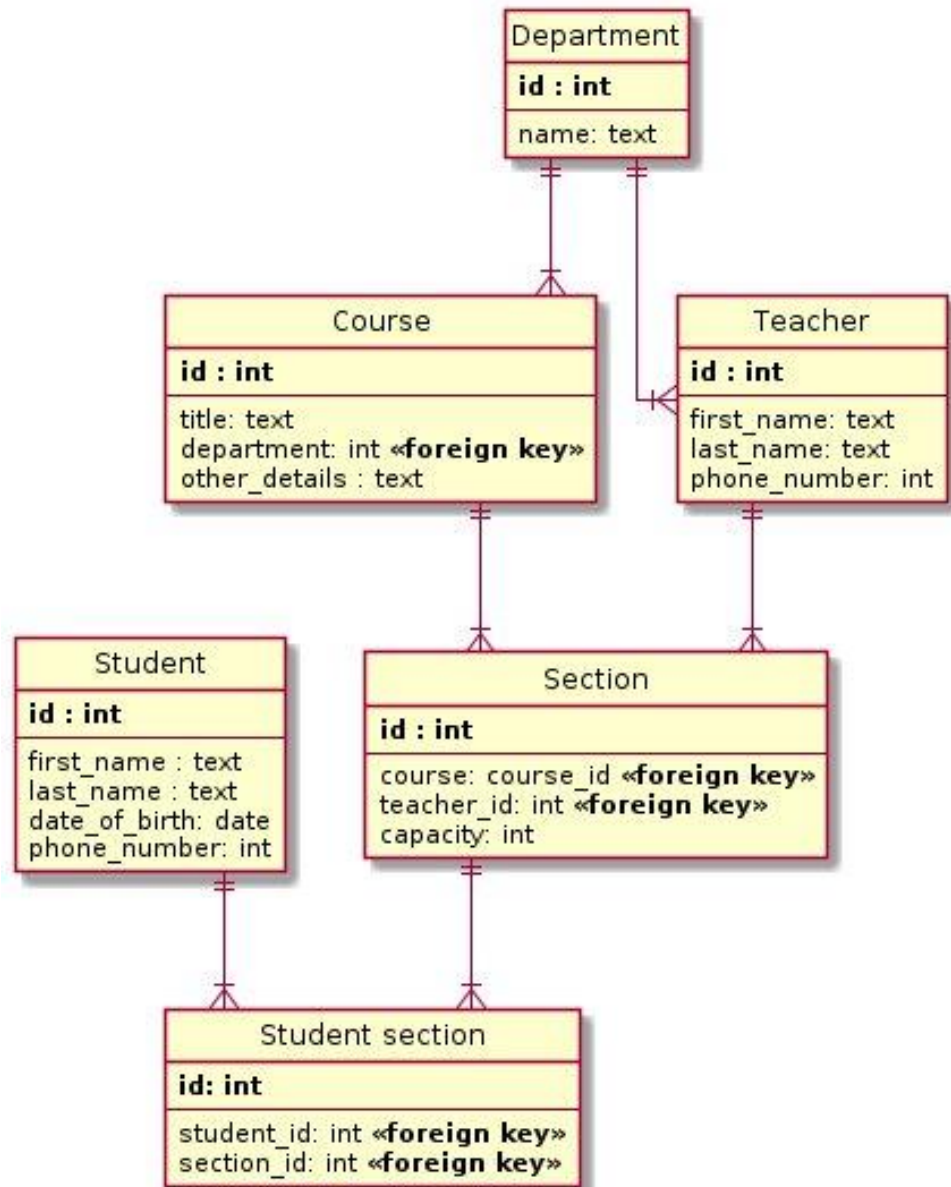


Рис. 1.2 Приклад побудови SQL баз даних

Наслідки:

- поле в таблиці, зване зовнішнім ключем, може містити посилання на стовпці в інших таблицях, що дозволяє їх з'єднувати;

- високоорганізована структура і гнучкість робить реляційні БД потужними і такими, що адаптуються до різних типів даних;
- для доступу до даних використовується мова структурованих запитів (SQL);
- надійний вибір для багатьох додатків.

Окремо виділяють NoSQL бази даних. NoSQL – група типів БД, що пропонують підходи, відмінні від стандартного реляційного шаблону. Говорячи NoSQL, мають на увазі або «не-SQL», або «не тільки SQL», щоб уточнити, що іноді допускається SQL-подібний запит.

5. Бази даних «ключ-значення»

У базах даних «ключ-значення» для зберігання інформації формується ключ і об'єкт даних, який потрібно зберегти. Наприклад, JSON-об'єкт, зображення або текст (рис. 1.3). Щоб зробити запит, відправляєте ключ і отримуєте blob-об'єкт.

key:	value
user_id:	f5badc33-5bd7-4b65-a737-b5304675f476
color:	blue
repetitions:	3
text:	hello world
data:	{ ... }

Рис. 1.3 Приклад побудови баз даних «ключ-значення»

Наслідки:

- сховища забезпечують швидкий доступ з незначними витратами;
- часто зберігають дані конфігурацій і інформацію про стан даних, представлених словниками або хешем;
- немає жорсткої схеми відносини між даними, тому в таких БД часто зберігають одночасно різні типи даних;
- розробник відповідає за визначення схеми іменування ключів і за те, щоб значення мало відповідний тип / формат.

6. Документні бази даних

Документні бази даних (також документоорієнтовані БД або сховища документів), спільно використовують базову семантику доступу і пошуку сховищ ключів і значень (рис. 1.4). Такі БД також використовують ключ для унікальної ідентифікації даних. Різниця між сховищами «ключ-значення» і документними БД полягає в тому, що замість зберігання blob-об'єктів, документоорієнтовані бази зберігають дані в структурованих форматах – JSON, BSON або XML.

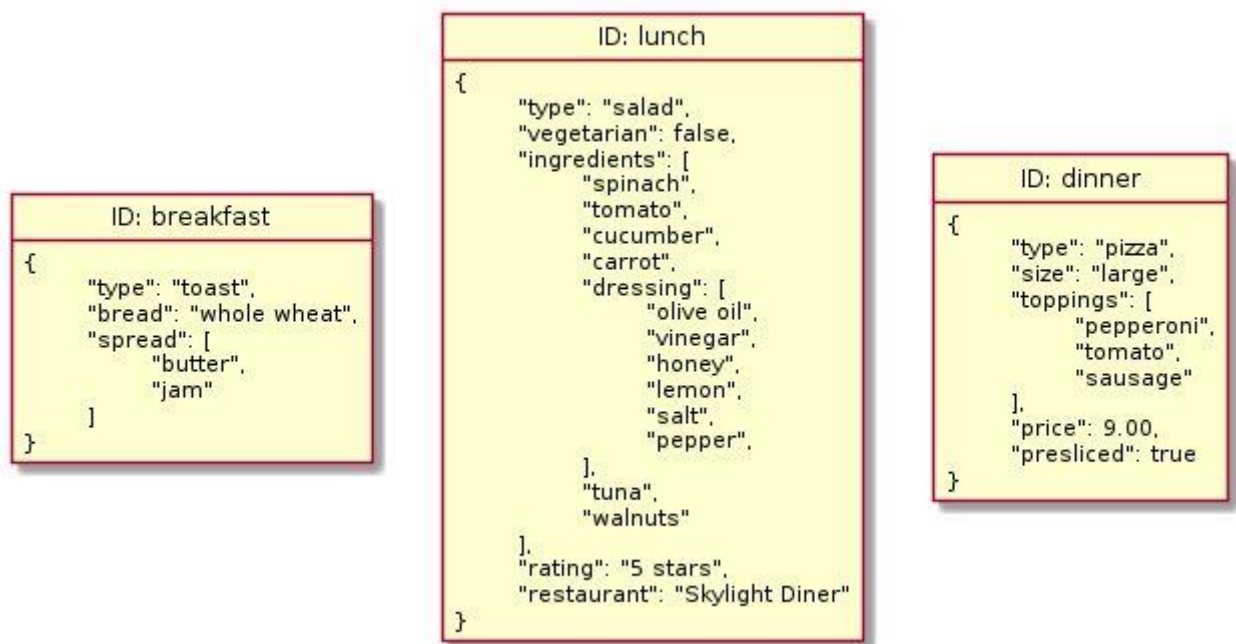


Рис. 1.4. Приклад побудови документних баз даних

Наслідки:

- база даних не виділяє окремий формат або схему;
- кожен документ може мати свою внутрішню структуру;
- документні БД є хорошим вибором для швидкої розробки;
- в будь-який момент можна змінювати властивості даних, не змінюючи структуру або самі дані.

7. Графові бази даних

Замість зіставлення зв'язків з таблицями і зовнішніми ключами, графові бази даних встановлюють зв'язки, використовуючи вузли, ребра і

властивості. Графові бази представляють дані у вигляді окремих вузлів, які можуть мати будь-яку кількість пов'язаних з ними властивостей (рис. 1.5).

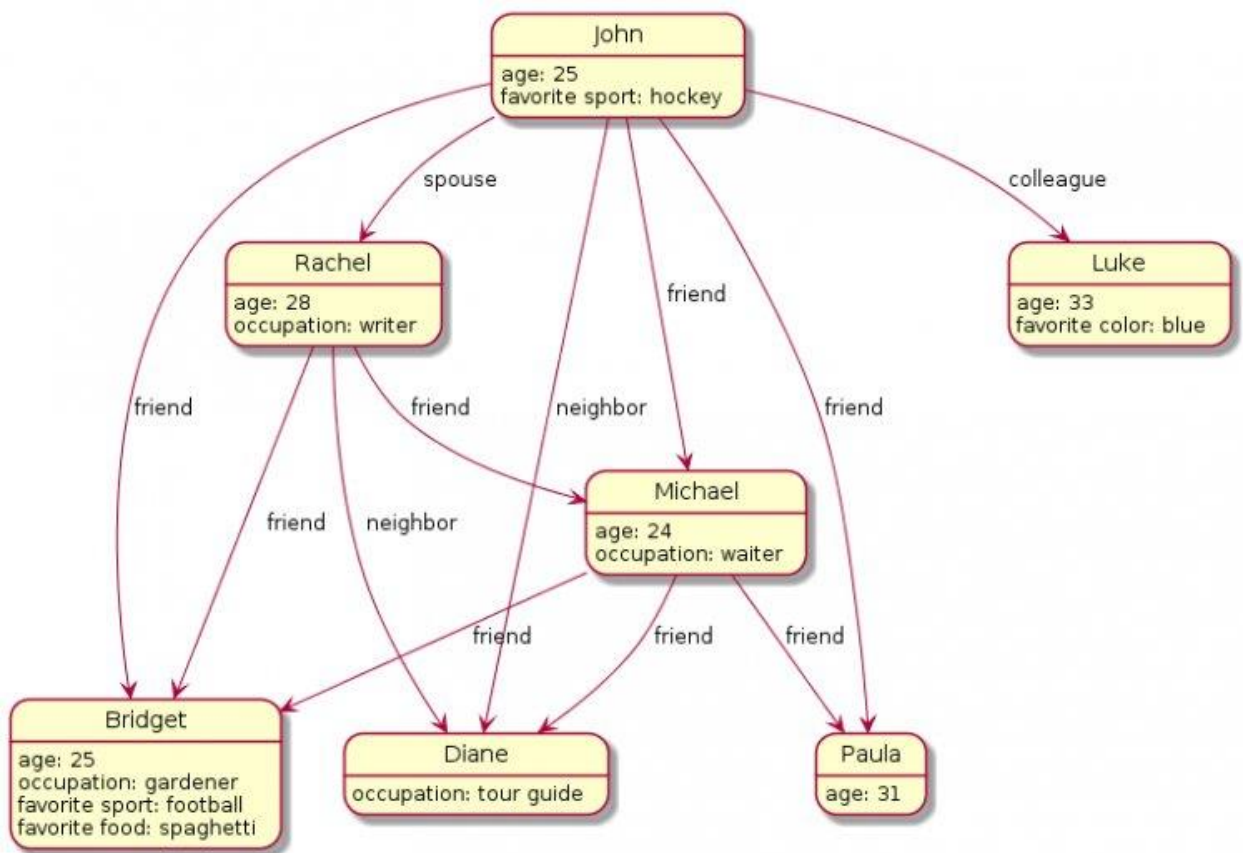


Рис. 1.5. Приклад побудови графових баз даних

Наслідки:

- виглядають аналогічно мережевим;
- фокусуються на зв'язках між елементами;
- явно відображають зв'язки між типами даних;
- не вимагають покрокового обходу для переміщення між елементами;
- немає обмежень в типах зв'язків.

8. Стовбчикові бази даних

Стовбчикові бази даних (також нереляційні колоночні сховища або бази даних з широкими стовпцями) належать до сімейства NoSQL БД, але зовні схожі на реляційні БД. Як і реляційні, стовпчикові БД зберігають дані,

використовуючи рядки і стовпці, але з іншим зв'язком між елементами (рис. 1.6).

У реляційних БД всі рядки повинні відповідати фіксованій схемою. Схема визначає, які стовпчики будуть в таблиці, типи даних та інші критерії. У стовпчикових базах замість таблиць є структури - «стовпчик сімейства». Сімейства містять рядки, кожна з яких визначає власний формат. Рядок складається з унікального ідентифікатора, що використовується для пошуку, за яким слідують набори імен та значень стовпців.

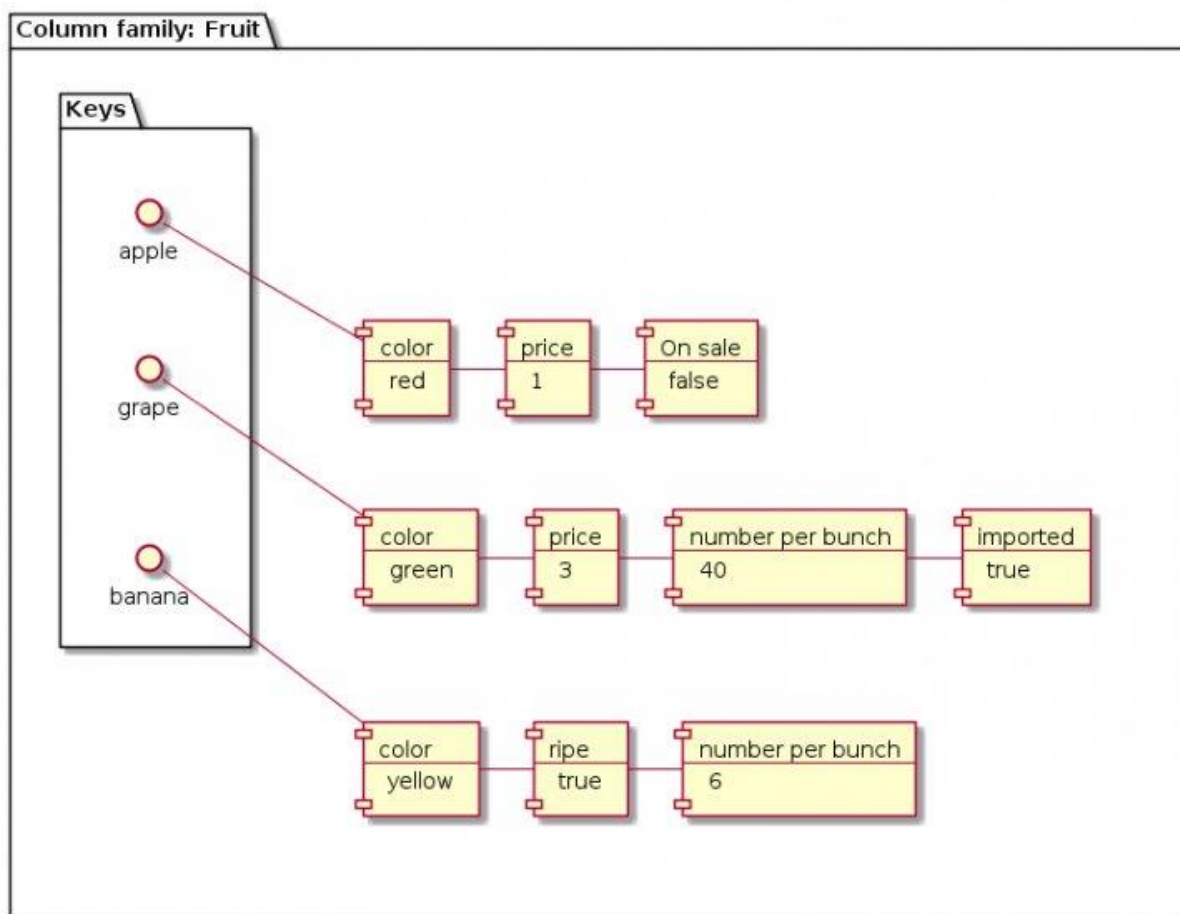


Рис. 1.6. Приклад побудови стовпчикових баз даних

Наслідки:

- БД зручні при роботі з додатками, що вимагають високої продуктивності;
- дані та метадані доступні по одному ідентифікатору;

- гарантоване розміщення всіх даних з рядка в одному кластері, що спрощує сегментацію і масштабування даних.

9. Бази даних часових рядів

Бази даних часових рядів створені для збору і управління елементами, що змінюються з плином часу. Більшість таких БД організовані в структури, які записують значення для одного елемента. Наприклад, можна створити таблицю для відстеження температури процесора (рис. 1.7). У середині кожне значення буде складатися з тимчасової мітки і показника температури. У таблиці може бути декілька метрик.

Time	CPU Temp	System Load	Memory Usage %
2019-10-31T03:48:05+00:00	37	0.85	92
2019-10-31T03:48:10+00:00	42	0.87	90
2019-10-31T03:48:15+00:00	33	0.74	87
2019-10-31T03:48:20+00:00	34	0.72	77
2019-10-31T03:48:25+00:00	40	0.88	81
2019-10-31T03:48:30+00:00	42	0.89	82
2019-10-31T03:48:35+00:00	41	0.88	82

Рис. 1.7. Приклад побудови баз даних часових рядів

Наслідки:

- орієнтовані на запис;
- призначені для обробки постійного потоку вхідних даних;
- продуктивність залежить від кількості відслідковуваних елементів, інтервалу опитування між записом нових значень і фактичного корисного навантаження даних.

NewSQL і багатомодельні БД є різними типами баз даних, але вирішують одну групу проблем, викликаних полярними підходами SQL або NoSQL-стратегії. Чому б не об'єднати переваги обох груп?

10. NewSQL бази даних

NewSQL бази даних успадковують реляційну структуру і семантику, але побудовані з використанням більш сучасних, масштабованих конструкцій. Мета – забезпечити більшу масштабованість, ніж реляційні БД, і більш високі гарантії узгодженості, ніж в NoSQL. Компроміс між

узгодженістю і доступністю є фундаментальною проблемою розподілених баз даних, описаних теоремою CAP.

Наслідки:

- можливість горизонтального масштабування;
- висока доступність;
- велика продуктивність і реплікація;
- невеликий функціонал і гнучкість;
- чимале споживання ресурсів і необхідність спеціалізованих знань для роботи з базою даних.

11. Багатомодельні бази даних

Багатомодельні бази даних – бази, які поєднують функціональні можливості кількох видів БД. Переваги такого підходу очевидні - одна і та ж система може використовувати різні уявлення для різних типів даних.

Спільне розміщення даних з декількох типів БД в одній системі дозволяє виконувати нові операції, які в іншому випадку були б ускладнені або неможливі. Наприклад, багатомодельні бази можуть дозволити користувачам отримати доступ до даних, що зберігаються в різних типах БД, і управляти ними в рамках одного запиту, а також підтримують узгодженість даних при виконанні операцій, що змінюють інформацію відразу в декількох системах.

Наслідки:

- допомагають зменшити навантаження на СУБД;
- дозволяють розширюватися до нових моделей у міру зміни потреб без внесення змін до базової інфраструктури;
- забезпечують безперервний доступ і простий розподіл даних;
- мають лінійну масштабованість і прості для розробки.

Загалом, зміна типів даних, що зберігаються, вимоги до швидкості і продуктивності привели до триваючого розширення типів баз даних (рис. 1.8). При цьому кожен з них продовжує бути потрібним у своїй ніші, де взаємозв'язки між даними асоціюються з певною схемою будови бази даних.

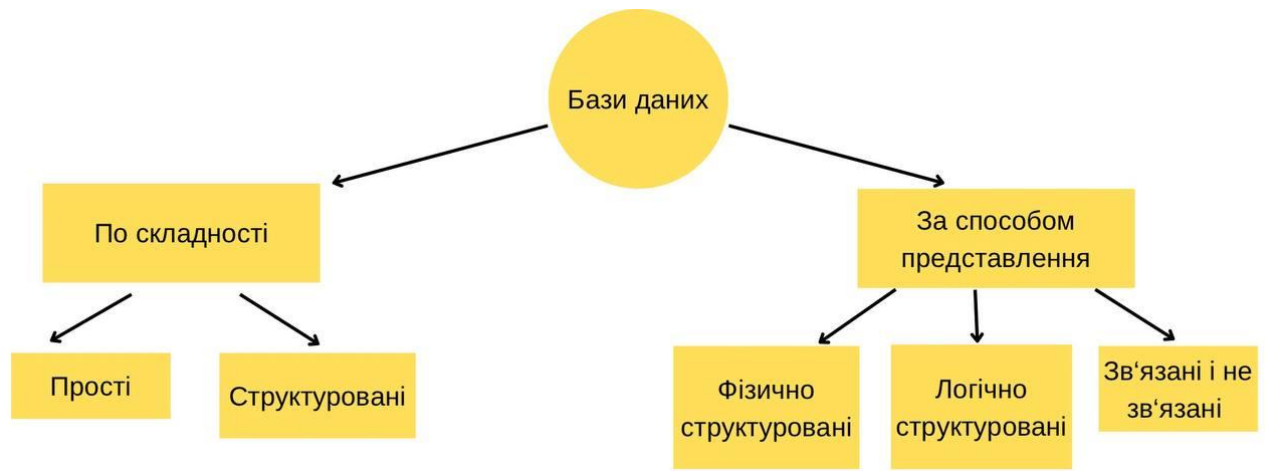


Рис.1.8. Класифікація структур даних

Класифікація структур даних може бути виконана за різними ознаками.

1) По складності: прості і інтегровані.

Прості (базові, примітивні) структури – це такі, які не можуть бути розподілені на складові частини.

Структуровані (інтегровані, композитні, складні) – такі структури даних, складовими частинами яких є інші структури даних – прості, у свою чергу, інтегровані. Інтегровані структури даних конструюються програмістом.

2) По способу представлення: фізична і логічна.

Фізична структура даних – це спосіб фізичного подання даних у пам'яті комп'ютера.

Логічна або абстрактна структура – це розгляд структури даних без урахування його подання в машинної пам'яті.

В загальному випадку між логічної і відповідної їй фізичної структурами існує розбіжності, ступінь якого залежить від самої структури та особливостей того середовища, в якому вона повинна бути відображена. Внаслідок цього існують розбіжності процедури, які здійснюють відображення логічної структури в фізичну, і, навпаки, фізичної структури в логічний.

3). За наявності зв'язків між елементами даних: незв'язні і зв'язні.

Незв'язні структури характеризуються відсутністю зв'язків між елементами структури.

Зв'язні структури характеризуються наявністю зв'язку. Прикладами незв'язних структур є вектори, масиви, рядки, стеки, черги; приклади зв'язних структур - зв'язні списки.

4). За мінливості: статичні, полустатичні, динамічні.

Мінливість, тобто зміна числа елементів і (чи) зв'язків між елементами структури.

Статичні – до цієї групи відносять масиви, безлічі, запису, таблиці.

Полустатичні – це стеки, черзі, деки, дерева.

Динамічні – лінійні та розгалужені зв'язні списки, графи, дерева.

5). За характером впорядкованості елементів у структурі: лінійні і нелінійні.

Лінійні структури залежно від характеру взаємного розташування елементів пам'яті поділяють на структури з послідовним розподілом елементів пам'яті (вектори, рядки, масиви, стеки, черги) і структури з довільним зв'язковим розподілом елементів пам'яті (однозв'язні і двозв'язні лінійні списки).

Нелінійні структури - багатозв'язані списки, дерева, графи.

б). По виду пам'яті, що використовується для збереження даних: структури даних для оперативної і для зовнішньої пам'яті.

Структури даних для оперативної пам'яті - це дані, розміщені в статичної та динамічної пам'яті комп'ютера. Всі вищенаведені структури даних - це структури для оперативної пам'яті.

Структури даних для зовнішньої пам'яті називають файловими структурами або файлами. Прикладами файлових структур є послідовні файли, файли, організовані розділами, - дерева.

Тип даних однозначно визначає:

а) структуру зберігання даних зазначеного типу, тобто розподіл пам'яті й подання даних у ній, з одного боку, і інтерпретацію двійкового подання, з іншого;

б) допустимі значення, які може мати об'єкт описуваного типу;

в) допустимі операції, які можуть бути застосовні до об'єкту описуваного типу.

1.2. Особливості управління базами даних

База даних є сховищем упорядкованої сукупності даних. Для роботи з БД використовують системи управління базами даних (СКБД).

Система управління базами даних – це комплекс програм, що забезпечує введення, зберігання, пошук, опрацювання даних у базі даних.

До популярних СКБД належать MySQL, PostgreSQL, Microsoft SQL Server, Oracle, Sybase, Interbase, Firebird і IBM Db2.

Залежно від моделі даних, яка використовується в СКБД, їх поділяють на ієрархічні, мережеві, реляційні, об'єктно-реляційні тощо.

Прикладом СКБД є Microsoft Office Access. Вона працює з об'єктами (таблиці, запити, форми, звіти тощо), які найчастіше зберігаються в одному файлі з розширенням .accdb.

Створення бази даних слід починати з її проектування (розробки). У результаті проектування має бути визначена структура бази, тобто склад таблиць, їхня структура та логічні зв'язки. Структура реляційної таблиці визначається складом стовпців, їхньою послідовністю, типом даних кожного стовпця та їхнім розміром, а також ключем таблиці. Процес проектування можна здійснювати двома підходами. За першого підходу спочатку визначають основні задачі, для розв'язання яких створюється база, та потреби цих задач у даних. За другого підходу визначають предметну область (сферу), здійснюють аналіз її даних і встановлюють типові об'єкти предметної області. Найбільш раціональним підходом проектування бази даних є поєднання обох підходів.

Види даних в базах даних

Об'єкти	Опис
Таблиці	Зберігають дані, що відповідають назві таблиці. Ці дані розбиті на окремі факти, що зберігаються в окремих полях. Наприклад, у таблиці Працівники у полі Прізвище зберігаються прізвища, у полі Номер - табельні номери. Сукупність фактів про одного працівника складає один запис
Запити	Використовується для опрацювання даних із однієї або кількох зв'язаних таблиць, пошуку даних за певними умовами та для обчислення підсумкових значень. Деякі запити використовуються як джерело даних для форм і звітів, деякі - для проведення різноманітних операцій над даними. Результати виконання цих операцій можна переглянути в режимі таблиці запиту
Форми	Надають зручні засоби для роботи з даними таблиць і запитів
Звіти	Призначені для виведення даних на друк, проте їх можна переглядати на екрані, зокрема для того, щоб перед друком оцінити, як виглядатиме документ і за потреби внести зміни

Зазвичай з базами даних працюють дві категорії користувачів. Перша категорія - проектувальники. Процес проектування бази даних поділяється на етапи, кожний з яких передбачає виконання певних дій. Перший етап - розробка інформаційно-логічної моделі даних предметної області, який базується на описі предметної області, отриманому в результаті її обстеження. На цьому етапі спочатку визначають склад і структуру даних предметної області, які мають міститись у базі даних та забезпечувати виконання запитів, задач і застосувань користувача. Ці дані мають форму реквізитів, що містяться в різних документах – джерелах завантаження бази даних. Аналіз виявлених даних дозволить визначити функціональні залежності реквізитів, які використовують для виділення інформаційних об'єктів, що відповідають вимогам нормалізації даних. Подальше визначення

структурних зв'язків між об'єктами дозволяє побудувати інформаційно-логічну модель.

Другий етап - визначення логічної структури бази даних. Для реляційної бази даних цей етап є значною мірою формальним, оскільки інформаційно-логічна модель відображається в структуру реляційної бази даних адекватно.

Наступний етап - конструювання таблиць бази даних, який здійснюється засобами СУБД, та узгодження їх із замовником. Структура таблиць бази даних задається за допомогою засобів опису (конструювання) таблиць у СУБД із цілковитою відповідністю інформаційним об'єктам. Крім таблиць, проектувальники розробляють й інші об'єкти бази даних, які призначені, з одного боку, для автоматизації роботи з базою, а з іншого – для обмеження функціональних можливостей роботи з базою (безпека бази даних). Після формування структури таблиць база даних може наповнюватись даними з документів-джерел. Проектувальники, як правило, не наповнюють базу конкретними даними (оскільки замовник може вважати дані конфіденційними і не надавати стороннім особам). Винятком є експериментальне наповнення модельними даними на етапі відлагодження об'єктів бази.

Друга категорія – користувачі, які працюють із базами даних. Вони отримують вхідну базу даних від проектувальників, наповнюють її та обслуговують. Користувачі не мають засобів доступу до управління структурою бази, вони мають доступ тільки до даних, при цьому тільки до тих, робота з якими передбачена на їхньому конкретному робочому місці.

Проектування бази даних починається з вивчення технічного завдання на проектування бази даних, яке повинен надати замовник. Отже, бажано, щоб замовник володів відповідною термінологією і знав, принаймні в загальних рисах, технічні можливості основних СУБД. На жаль, на практиці ці побажання виконуються не завжди. Тому зазвичай розробники використовують такі підходи: демонструють замовникові роботу аналогічної

бази даних, після чого узгоджують специфікацію відмінностей; якщо аналога немає, з'ясовують коло задач і вимог замовника, після чого допомагають йому підготувати технічне завдання. Під час підготовки технічного завдання складають: перелік вхідних даних, з якими працює замовник; перелік вихідних даних, потрібних замовникові для управління структурою свого підприємства; перелік вихідних даних, які не є необхідними для замовника, але які він повинен надати іншим організаціям (у вищестоящі структури, в органи статистики, інші адміністративні і контрольні організації).

Визначивши основну частину даних, які замовник використовує, розпочинають розробку структури бази, тобто структури її основних таблиць.

1. Робота починається з визначення генерального переліку полів, який може нараховувати десятки і сотні позицій.

2. Відповідно до типу даних, що розміщуються в кожному полі, визначають тип кожного поля.

3. Розподіляють поля генерального списку по базових таблицях. На першому етапі розподіл здійснюють за функціональною ознакою. Мета - забезпечити одноразове введення даних в одну таблицю по можливості в рамках одного підрозділу, або (ще краще) - на одному робочому місці. На другому етапі розподілу полів здійснюють нормалізацію даних з метою вилучення повторів даних у таблицях бази даних.

4. Для кожної таблиці визначають ключове поле. Ключовим вибирають поле, дані в якому повторюватись не можуть. Наприклад, для таблиці даних про студентів таким полем може бути індивідуальний шифр студента. Для таблиць, у яких міститься розклад занять, такого поля можна і не знайти, але його можна створити штучно комбінуванням полів "Час заняття" і "Номер аудиторії". Ця комбінація унікальна, оскільки в певній аудиторії в певний час назагал не проводять двох різних занять. Якщо ж у таблиці взагалі немає полів, які можна було б використовувати як ключові, завжди можна ввести

додаткове поле типу лічильник - воно за визначенням не може містити дані, що повторюються.

5. На наступному етапі визначають зв'язки між таблицями (схему даних). Зв'язки між таблицями організуються на основі спільного поля, причому в одній із таблиць воно обов'язково має бути ключовим. Тобто на стороні "один" має бути ключове поле, яке не повторюється, значення на стороні "багато" можуть повторюватися. 6. "Паперовий" етап роботи над технічними пропозиціями закінчується розробкою схеми даних. Цю схему слід узгодити із замовником, після чого розпочати безпосереднє створення бази даних. Слід пам'ятати, що в ході розробки проекту замовникові неодмінно будуть надходити нові ідеї. Можливість гнучкого використання його побажань суттєво залежить від кваліфікації розробника бази даних. Якщо схема даних складена правильно, підключити до бази нові таблиці неважко. Якщо структура бази нераціональна, розробник може наштотхнутись на суттєві труднощі і дійти суперечності із замовником. Суперечка виконавця із замовником завжди свідчить про недостатню кваліфікацію виконавця. На цьому етапі завершується попереднє проектування бази даних, і на наступному етапі починається її безпосередня розробка (впровадження).

1.3 Технологія використання СУБД

СУБД є програмним продуктом, що постачається як пакет прикладних програм, який має бути встановлений (інстальований) на комп'ютер з урахуванням його конфігурації, ресурсів та операційної системи, а також вимог до набору функцій.

СУБД є основою створення практичних функціональних програм користувача для різних предметних областей. На вибір СУБД для функціональних програм впливає багато факторів, до яких належать: вимоги до продуктивності при обробці даних; наявне технічне і базове програмне забезпечення, їх конфігурація, оперативна і дискова пам'ять; тип моделі

даних, що підтримується, специфіка предметної області, топологія інформаційно-логічної моделі; потреби функціональних програм користувача, що розробляються; наявність у СУБД необхідних функціональних засобів; рівень кваліфікації користувачів і наявність у СУБД діалогових засобів розробки та роботи із СУБД.

Після встановлення СУБД можна здійснювати створення БД, у тому числі – задавати структуру БД, здійснювати введення даних, а також виконувати будь-які дії, що передбачені функціональними можливостями СУБД. Слід зазначити, що сучасні СУБД для персональних комп'ютерів (ПК) мають достатню гнучкість. Це дозволяє на перших етапах розробки функціональних програм користувача розпочати створення окремих частин БД. Така БД у міру поглиблення розробки може легко модифікуватися. Таким чином полегшується прискорене засвоєння персоналом технології роботи з БД, вивчення можливостей СУБД і поетапне її впровадження.

Розробка функціональних програм користувача на основі СУБД передбачає підготовку рішень щодо структури БД. Ці рішення безпосередньо пов'язані з позамашиною сферою – з описом позамашиної інформаційної бази, її документів, що містять потрібну інформацію, а також з постановкою й алгоритмізацією задач із обробки цієї інформації.

На початковому етапі розробки структури БД доцільна побудова інформаційно-логічної моделі, що відображає логічну структуру інформації предметної області. Така модель, яка відповідає вимогам нормалізації даних, є основою створення реляційних баз даних.

Найбільш розповсюдженими є Microsoft Access, FoxPro, Paradox, Clipper.

Відповідно до розробленої структури бази даних здійснюється її створення засобами СУБД на машинному носії та введення в експлуатацію. Забезпечити процеси створення БД та її експлуатації можна тільки на основі знання можливостей інструментальних засобів СУБД. При цьому слід керуватися рекомендаціями з технології використання засобів СУБД. Така

технологія має визначати всі необхідні процеси, включаючи початкове введення, завантаження БД і контроль за даними, виконання операцій із внесення змін, реалізацію запитів для отримання потрібних довідок, відновлення БД тощо. Один із найважливіших етапів цієї технології - підготовка екранних форм введення/виведення для завантаження інформації з документів позамашинної сфери в базу даних, редагування даних та їх перегляду.

Поповнення, вилучення, зміна і вибірка даних здійснюються за допомогою мови запитів, вбудованої алгоритмічної мови та інших засобів СУБД. Реалізація запитів забезпечується або діалоговою системою команд з меню, або запитамі за прикладом QBE (Query By Example). У першому випадку окремий запит виконується однією або кількома командами мови СУБД. Послідовність команд мови СУБД утворює програму - командний файл. У другому - для виконання запиту користувач вибирає послідовно один чи кілька пунктів меню або вказує в запиті приклад (зразок), за яким створюється запит, а також у разі потреби - умови вибору й операції обчислення, які необхідно виконувати з даними (СУБД Paradox, Access тощо). Послідовність команд меню і запитів може запам'ятати програма-макрос та в подальшому виконати її так само, як командний файл.

У СУБД може використовуватись базова мова або одна з універсальних алгоритмічних мов (C, Pascal тощо). Прикладна програма, що написана універсальною алгоритмічною мовою, може ініціювати команди СУБД. У СУБД з базовою мовою застосовується власна алгоритмічна мова, що дозволяє, крім операцій маніпулювання даними, виконувати різні обчислення й обробку даних. Стандартною реляційною мовою запитів є мова структурованих запитів SQL (Structured Queries Language).

На практиці доступ до бази даних найчастіше має одна програма (один користувач). Однак бувають ситуації, коли одночасно (паралельно) виконується кілька програм або кілька прогонів однієї і тієї ж програми. У системі продажу проїзних квитків, наприклад, продавати квитки і змінювати

таким чином кількість вільних місць можуть одночасно кілька агентів. Тут слід потурбуватися про правила, які регулюють доступ до бази даних двох або більше програм, в іншому разі не виключена можливість продажу квитків на одне і те саме місце двом пасажиром. Таким чином, не слід допускати паралельного виконання двох процесів, які читають і змінюють значення одного і того ж об'єкта. Аналогічний приклад можна навести з платіжної технології щодо переказу грошових коштів з одного рахунку на інший.

Іншим прикладом може бути база даних перепису населення. До такої бази даних одночасно можуть звертатися кілька користувачів, оскільки при цьому жоден з них не змінює даних, користувачі лише читають їх. У схожих ситуаціях, коли здійснюється лише читання, для економії часу бажано максимально використовувати паралельні операції. У зв'язку з цим технологія баз даних вимагає опису доступу до її елементів (доцільність доступу, що дозволяє виконувати паралельні операції лише в тому разі, коли вони зберігають цілісність бази даних, і заборона паралельності операцій, коли вони порушують цілісність бази даних).

Одним зі способів уникнення паралельного читання і зміни одного і того ж об'єкта є блокування окремих елементів бази. База даних розбивається на окремі частини, які можна блокувати. Блокуючи певну частину, транзакція (разовий «прогін» програми) може перешкоджати доступу інших транзакцій до цієї частини бази до того моменту, доки вона не розблокує її. Функціональний компонент СУБД (управління блокуванням) призначає і реєструє блокування, а також відіграє роль арбітра між двома (і більше) запитами на блокування однієї і тієї ж частини бази даних. Елементами блокування в реляційній моделі даних можуть бути окремі компоненти кортежів, окремі кортежі, блок кортежів і навіть відношення.

Щоб переконатися в необхідності блокування елементів бази даних, розглянемо дві транзакції – T1, і T2. Кожна з них здійснює доступ до

елемента A і збільшує його значення на одиницю. Обидві транзакції є «прогоном» програми P :

$P: \text{Read } A; A := A+1; \text{Write } A.$

Значення A міститься в базі даних. Програма P читає (*Read*) значення A у свій робочий простір, додає до нього одиницю і записує (*Write*) результат.

A у базу даних (табл. 1.2) наведено ці транзакції і відповідні значення A в базі даних на кожному кроці виконання транзакцій.

Таблиця 1.2

Транзакції, що демонструють необхідність блокування елемента A

Значення A в базі даних	7	7	7	7	8	8
Транзакція T_1						
Транзакція T_2	Read	Read	$A:=A+1$	$A:=A+1$	Write	Write
Значення A в робочому просторі T_1	A	A	1		A	A
Значення A в робочому просторі T_2	7	7	7	8	8	8

Легко помітити, що хоч кожна транзакція і додає до елемента A одиницю, його значення зростає тільки на одиницю. Це викликає серйозну проблему, якщо, наприклад, врахувати, що A – кількість проданих квитків на певний авіарейс, тобто в цьому разі на одне місце може бути продано два квитки.

Вирішення цієї проблеми – у забезпеченні блокування для елемента A . Перед тим як прочитати A , транзакція T_i має його блокувати, що дасть змогу запобігти доступу другої транзакції до нього доти, доки T_i не завершить

роботу з А. Програму Р з блокуванням елемента А можна представити таким чином:

P: Lock A; Read A; A: = A + 1; Write A; Unlock A.

Якщо Т1 і Т2 – два виконання програми Р і першою починається Т1, то вона починається з блокування А. Система надає це блокування з умовою, що інша транзакція не заблокувала А, тоді транзакція Т1 (і лише вона одна) отримує доступ до А.

Якщо Т2 починається до завершення Т1 то як тільки Т2 спробує виконати команду Блокувати А (Lock А), система змусить її чекати до виконання першою транзакцією команди Розблокувати А (Unlock А). Тільки після цього система дозволить виконання транзакції Т2. Таким чином, аномалія не може мати місця, оскільки одна із транзакцій, Т1 або Т2, буде виконана повністю перед тим, як почнеться друга, а їх спільним результатом буде збільшення А на два.

До цього часу передбачалося, що кожна транзакція виконується вдало аж до свого логічного завершення. На практиці, однак, різні обставини можуть перешкодити її завершенню:

- з тих чи інших причин, що пов'язані з обладнанням або програмним забезпеченням, може відбутися відмова системи, у цьому разі всі активні транзакції не спроможні завершитися. Відмови системи породжують серйозні проблеми - потрібно не лише знайти множину транзакцій, роботу яких треба “анулювати”, щоб повернути систему в попередній стан, а й переконатися в тому, що є якийсь спосіб відновлення такого стану;
- виконання транзакції може бути примусово закінчено ще до її завершення (ділення на нуль тощо). У зв'язку з цим слід передбачати періодичне копіювання бази даних, службова програма для копіювання має сама бути транзакцією з блокуванням даних. Слід також зберігати на диску історію (журнал) усіх змін у базі даних. Записи журналу, як правило, містять: унікальний номер транзакції, що зумовила зміни в базі даних; попередні значення елементів бази; нові значення елементів тощо.

Базу даних можна уявити як людиноорієнтовану та комп'ютероорієнтовану модель даних, кожна з яких визначає своє середовище збереження даних. Людиноорієнтована модель бази даних відображається в паперових документах і в пам'яті людини, а комп'ютероорієнтована – на фізичних носіях ЕОМ. Кожний рівень даних характеризується своїм описом (сприйняттям) елементів даних – зовнішній (інфологічний), концептуальний (дatalogічний) і внутрішній (фізичний) рівень. Ці рівні формують трирівневу архітектуру баз даних.

Опис, виконаний із використанням природної мови, математичних формул, таблиць, графіків та інших засобів, зрозумілий людям, які працюють над проектуванням бази даних, називають інфологічною моделлю даних. Основними конструктивними елементами інфологічної моделі є сутності, зв'язки між ними та їхні властивості. Сутність – це будь-який об'єкт, інформацію про який потрібно зберігати в базі даних. Сутностями можуть бути товари, виробники товарів, банківські рахунки тощо. Властивість – іменована характеристика сутності (поточний рахунок, фірма «Світоч» тощо). Для прискорення пошуку конкретних атрибутів бази даних використовують ключі. Ключ – мінімальний набір атрибутів, за значенням яких можна однозначно знайти необхідний примірник сутності. Наприклад, для сутності Розклад_занять ключем є атрибут №_групи або набір: Дисципліна, Час і Лектор.

Одна з основних вимог до організації бази даних – це забезпечення можливостей знаходження одних інформаційних об'єктів за значенням інших, для цього необхідно встановити між ними певні зв'язки. Оскільки в реальних базах даних нерідко зберігаються сотні і навіть тисячі сутностей, то теоретично між ними може бути встановлено більше, ніж мільйон зв'язків. Наявність такої множини зв'язків і визначає складність інфологічних моделей.

Під час побудови інфологічних моделей, як правило, використовують мову ER-діаграм (від англ. Entity-Relationship – сутність-зв'язок). У них

сутності відображаються прямокутниками, асоціації – ромбами або шестикутниками, атрибути – овалами, а зв'язки між ними – ненаправленими ребрами, над якими можуть проставлятися ступені зв'язку (1 або буква, яка заміняє слово “багато”) і необхідні пояснення. Між двома сутностями, наприклад А і В, можливі чотири типи зв'язків.

Перший тип – зв'язок “один-до-одного” (1:1): у кожний момент часу кожному представникові сутності А відповідає 1 або 0 представників сутності В.

Другий тип – зв'язок “один-до-багатьох” (1:М): одному представникові сутності А відповідає 0,1 або кілька представників сутності В.

Між двома сутностями можливі зв'язки в обох напрямках, тому існують ще два типи зв'язків “багато-до-одного” (м:1) і “багато-до-багатьох” (м:п).

Побудова хорошої інфологічної моделі - досить складний процес. Досвід побудови таких моделей десятиліттями формували провідні спеціалісти у сфері обробки даних.

Інфологічна модель даних є основою для розробки концептуальної моделі даних (дatalogічної). Концептуальний рівень описує дані, які мають зберігатись на машинних носіях, а також зв'язки, які існують між ними, і містить логічну структуру всієї бази даних. На цьому рівні описуються типи даних, здійснюється їх нормалізація, визначається перелік файлів бази даних.

Висновки до першого розділу

В цьому розділі було детально вказані такі моменти як типи існуючих баз даних, їх основні відмінності і характеристики, розглянули схематичні побудови цих елементів.

Визначено, що існує 11 типів баз даних: прості структури даних, ієрархічні бази даних, мережеві бази даних, SQL бази даних, бази даних ключ-значення, документні бази даних, графові бази даних, стовбчикові бази

даних, бази даних часових рядів, NewSQL бази даних, багатомодельні бази даних.

Розрізняються бази даних за 6 критеріями (по складності, по способу представлення, за наявності зв'язків між елементами даних, за мінливості, за характером впорядкованості елементів у структурі, по виду пам'яті, що використовується для збереження даних).

РОЗДІЛ 2

МЕТОДИЧНІ ПІДХОДИ ДО ОПТИМІЗАЦІЇ БАЗ ДАНИХ

2.1. Методи масштабування реляційних баз даних

Масштабування реляційних баз даних відіграє вирішальну роль у забезпеченні оптимальної продуктивності та доступності високонавантажених систем. Існує різноманітні підходи до цього завдання, такі як реплікація Master/Slave та Master/Master, функціональне та горизонтальне партиціювання, шардинг та денормалізація. В даній статті розглянуть переваги та недоліки кожного з цих методів, а також будуть представлені сценарії, де вони виявляються найбільш ефективними.

Реплікація представляє собою процес створення та утримання копій баз даних на різних вузлах з метою забезпечення високої доступності, збереження даних та розподілу навантаження. Це базовий метод масштабування реляційних баз даних, існують дві основні схеми роботи реплікації: Master/Slave та Master/Master.

Master/Slave є одним з найбільш поширених та ефективних підходів до масштабування. В цьому випадку існує головний вузол (Master), який передає дані на одну або кілька реплік (Slave). Це також може бути організовано у формі дерева, де Slave передає дані іншим реплікам.

Зазвичай у підході Master/Slave всі зміни (insert/update/delete тощо) відбуваються на головному вузлі, звідки вони репліки отримують та зберігають. Отже, Master відповідає за запис та читання, тоді як Slave – виключно за читання.

Такий підхід може використовуватися для резервного копіювання, але в контексті масштабування – для розподілу навантаження. Маючи декілька копій БД, можна балансувати запити залежно від навантаження кожного вузла. Це допомагає, якщо ви хочете на головному вузлі мати найменшу затримку оновлення даних.

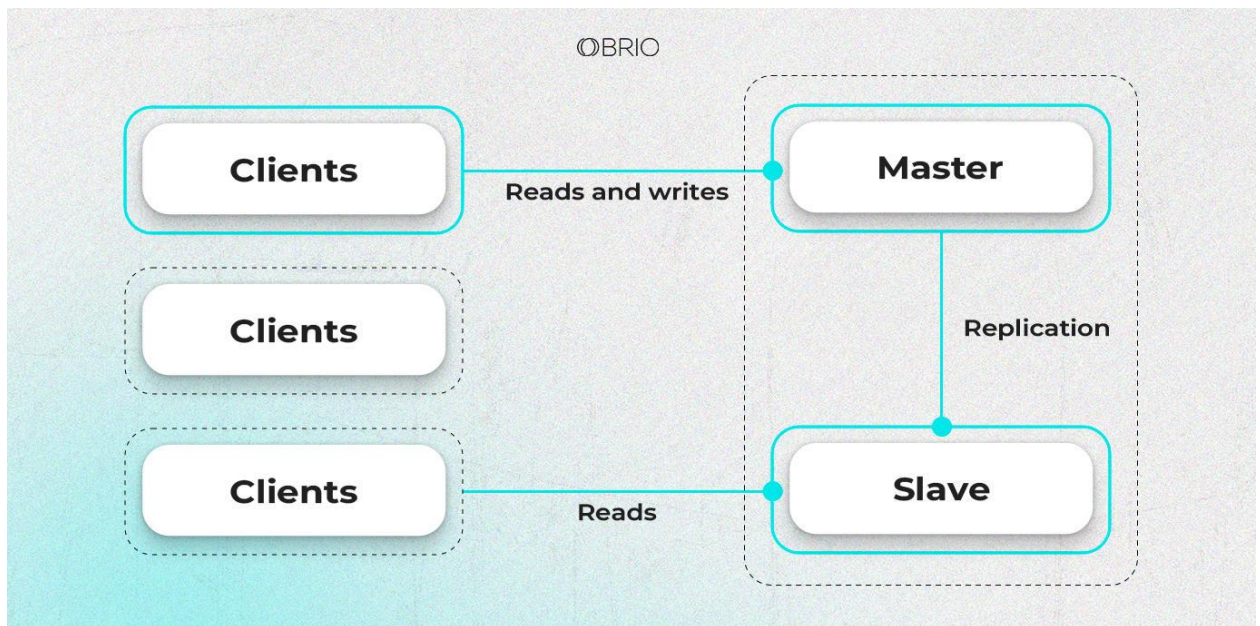


Рис. 2.1. Підході Master/Slave

Наприклад, є мобільний застосунок, яким щодня користуються мільйони людей. Через велику кількість запитів основна база даних матиме велике навантаження, що призведе до уповільнення часу відгуку сервера. У той час, коли більшість користувачів лише читають контент (наприклад, переглядають пости та коментарі), база даних опрацьовує як запити на читання, так і на запис. Використовуючи підхід Master/Slave, ви можете розділити операції читання на декілька серверів, що зменшить час завантаження контенту і покращить користувацький досвід.

Переваги методу:

чітке розділення ролей: один сервер БД виконує тільки функцію запису, а інші – тільки читання;

простота налаштування: порівняно з Master/Master реплікацією, Master/Slave є простішою в налаштуванні та управлінні;

відсутність конфліктів, оскільки лише один сервер БД (Master) опрацьовує записи;

оптимізація ресурсів: завдяки розподілу ролей, можна налаштувати сервер під конкретну задачу. Наприклад, для ролі Slave можна виділити більше RAM для кешування;

вузол Slave може служити для резервного копіювання і не впливатиме на інші сервери.

Недоліки методу:

ризик втрати даних, якщо Master вийде з ладу до того, як всі дані реплікуються на Slave;

додаткова складна логіка, щоби Slave зробити головним вузлом, якщо той вийде з ладу.

Підхід реплікації Master/Master корисний, коли вашій системі потрібно виконувати операції запису у великій кількості.

Наприклад, вебсайт, яким користуються люди з різних країн світу: створюють, редагують та видаляють контент. Щоби забезпечити найшвидший час відповіді в усіх регіонах, ви хочете розмістити сервери баз даних у різних географічних локаціях. Ваше завдання – забезпечити, щоб усі зміни були консистентні та синхронізовані між усіма вузлами в реальному часі.

Підхід Master/Master допоможе розв'язати цю проблему: зміни, внесені користувачем на одному сервері, автоматично реплікуються на всі інші, забезпечуючи консистентність даних у будь-якій точці світу.

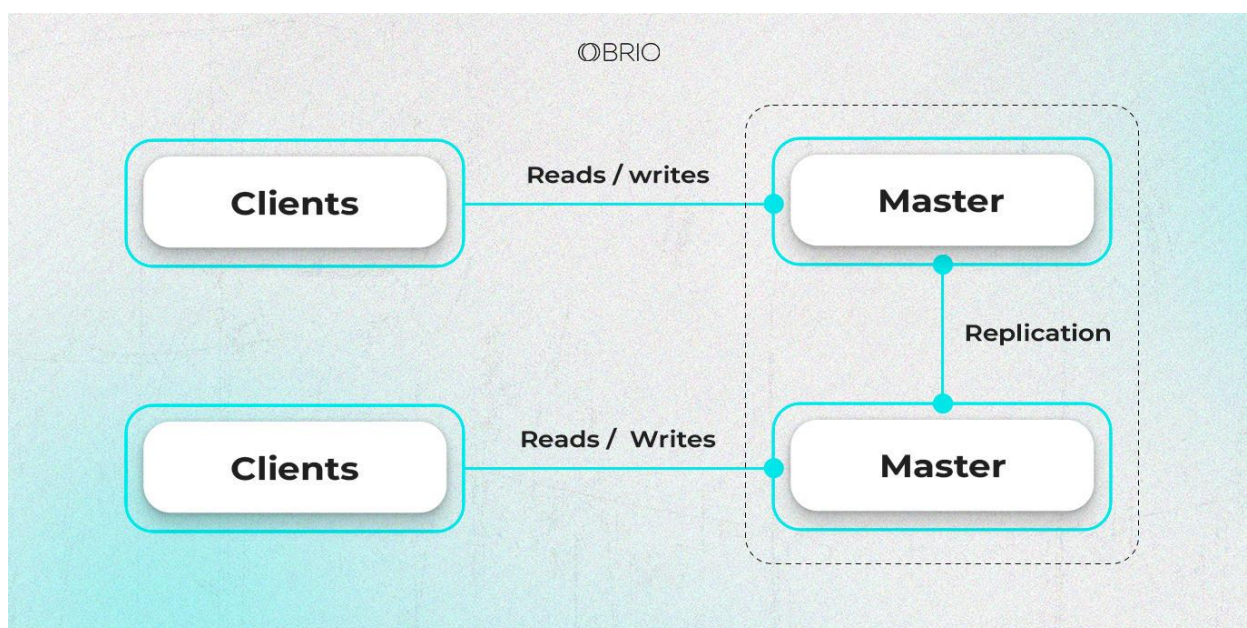


Рис. 2.2. Підхід Master/Master

Переваги методу:

доступність: якщо один із серверів БД виходить з ладу, система продовжує працювати без збою, адже його роль можуть виконувати інші репліки;

балансування: є можливість розділити навантаження всіх операцій;

географічна розподіленість: для кращого відгуку репліки можна розділити за регіонами.

Недоліки:

конфлікти записів: якщо два сервери одночасно вносять зміни в один рядок, виникає конфлікт, який потребує ручного втручання або специфічних стратегій вирішення;

Master/Master конфігурація може бути складнішою для налаштування та обслуговування порівняно з Master/Slave;

якщо використовується автоінкрементація для створення унікальних ID, потрібно налаштувати репліки, щоби вони не генерували однакові ідентифікатори;

безпека — оскільки декілька серверів БД можуть писати дані, можливі додаткові напрямки атак;

збільшений ризик втрати даних: проблема, що виникла на одному сервері, може швидко поширитися на інші через реплікацію.

Спільні недоліки реплікації для Master/Slave і Master/Master:

необхідність логіки в кодї, яка буде перенаправляти запити на потрібний сервер БД або додатковий рівень для балансування навантаження між репліками;

затримка оновлення даних: що більше ми маємо копій вузлів, то більше потрібно реплікувати даних, і це призводить до затримки;

затримка відповіді: якщо відбувається забагато запитів на запис, то репліки не зможуть швидко обслуговувати в тому числі запити читання;

збільшення інфраструктури: додаткові сервери потребують ресурсів на підтримку.

Функціональне партиціювання (Functional Partitioning/Database Federation) – це стратегія розділення бази даних на менші, спираючись на їхні функції.

Наприклад, застосунок, у якому користувачі можуть спілкуватися у чатах та здійснювати покупки. Якщо все розмістити на одному сервері БД, будь-яке збільшення навантаження на один компонент може вплинути на продуктивність всієї системи. Цю монолітну базу даних можна розділити на три маленьких: Chat, User, Purchase, – тоді кожна частина системи зможе масштабуватися незалежно від інших, враховуючи індивідуальні потреби та навантаження. Це покращить загальну продуктивність системи та забезпечить більшу стабільність: проблеми в одному компоненті не будуть безпосередньо впливати на інші.

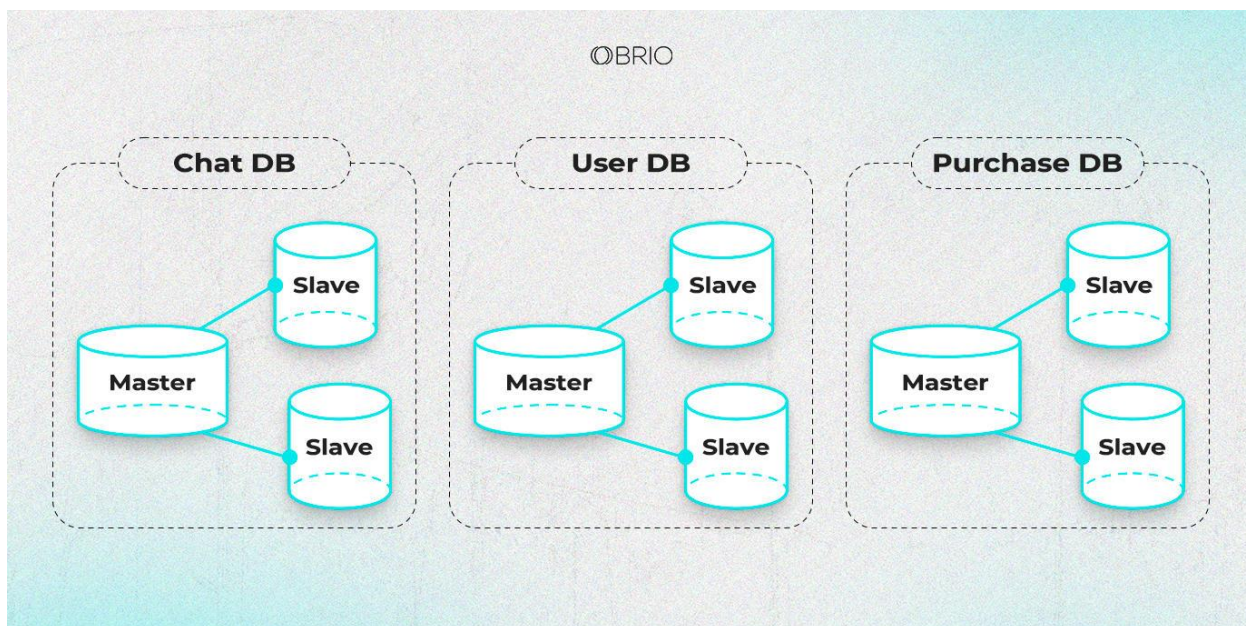


Рис. 2.3. Діаграма функціонального партиціювання

Переваги:

зменшення кількості запитів, що зменшує час реплікації та відгуку відповідно;

збільшення можливості кешування завдяки меншим розмірам бази даних;

- зменшення витрат на інфраструктуру;
- паралельне виконання запитів;
- збільшення ізольованості певного функціонала.

Недоліки:

- ускладнення логіки на рівні коду: потрібно правильно визначити, до якої бази робити запит;

- складність інфраструктури;

- об'єднання даних із різних БД стає непростим завданням;

- труднощі роботи з транзакціями.

Суть методу Шардинг – розділення однієї таблиці на декілька частин (шардів), кожна з яких зберігається на різних серверах.

Наприклад, є популярна соціальна мережа з мільйонами користувачів. Щодня вони генерують великі об'єми даних: публікації, коментарі, лайки тощо. Одна база даних не зможе ефективно обробляти цей обсяг інформації, що призведе до уповільнення та можливих відмов. Використовуючи шардинг, ви зменшуєте навантаження на окремий сервер, розподіляючи дані та обробку запитів між декількома сегментами. Кожен із них має менший об'єм даних для обробки, завдяки чому зменшується час відгуку. Коли з'являється необхідність, ми можете додавати нові шарди, у такий спосіб масштабуючи систему.

Якщо застосунок має глобальну аудиторію, можна розбити дані за регіонами, забезпечуючи швидкий відгук для користувачів із різних частин світу. Проте в такій реалізації може виникати нерівномірний розподіл навантаження, якщо певна група користувачів значно менше користується застосунком.

Переваги методу:

- зниження кількості запитів до сервера, часу реплікації та відповіді;

- збільшення даних в кеші через менші розміри бази даних;

- зменшення розміру індексу та часу запису;

- стабільність: якщо один шард вийде з ладу, інші працюватимуть;

паралельний доступ до даних.

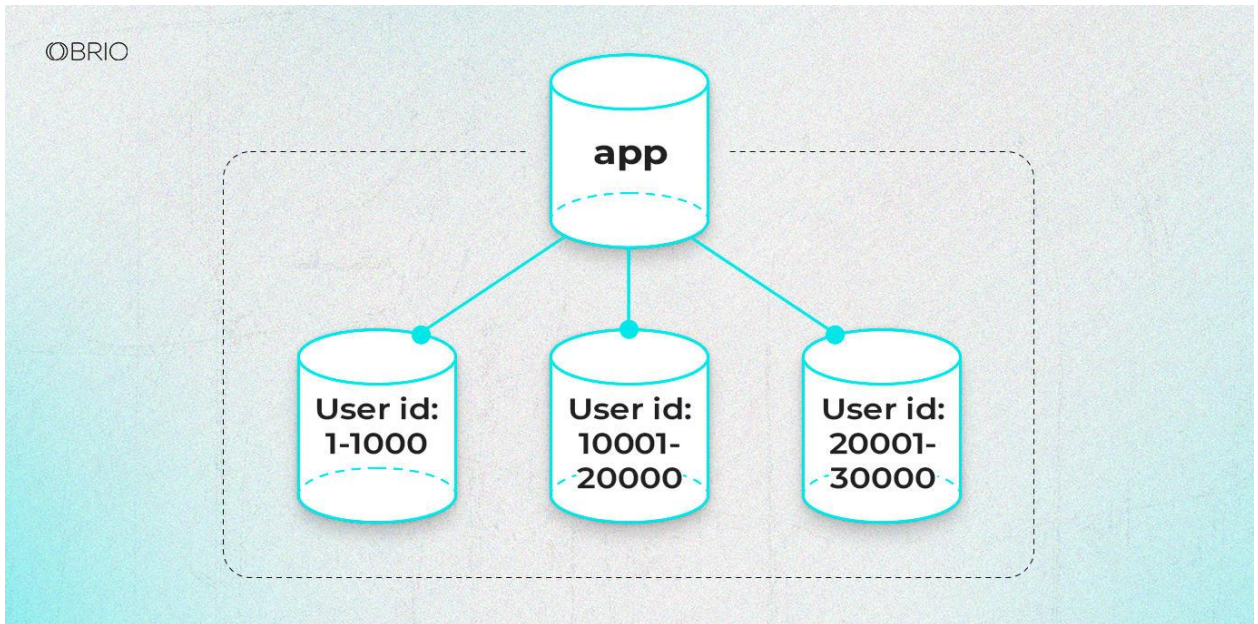


Рис. 2.4. Діаграма методу Шардинг

Недоліки:

ускладнення логіки на рівні коду, що вимагає навичок роботи з шардами;

складність об'єднання даних із різних сегментів;

потреба в збільшенні інфраструктури та її підтримці;

нерівномірне навантаження;

проблеми з транзакціями при роботі з даними з різних шардів.

Денормалізація передбачає дублювання даних або їхнє групування в БД, порушуючи правила нормалізації. Як це працює: ми дублюємо інформацію, щоб уникнути ресурсомістких операцій об'єднання (JOIN) та звернення до різних таблиць. У такий спосіб продуктивність запитів читання покращується шляхом запису.

Уявимо, що є застосунок, у якому користувачі можуть спілкуватися в чатах з експертами. Вам потрібно віддавати клієнту список чатів разом з іменами співбесідників. Операція JOIN між двома дуже великими таблицями буде займати чимало часу. Ситуація погіршиться, якщо таких запитів буде

багато. Щоби прискорити вибірку, можна перенести імена співбесідників до таблиці Chats, – тоді нам не потрібно робити JOIN.

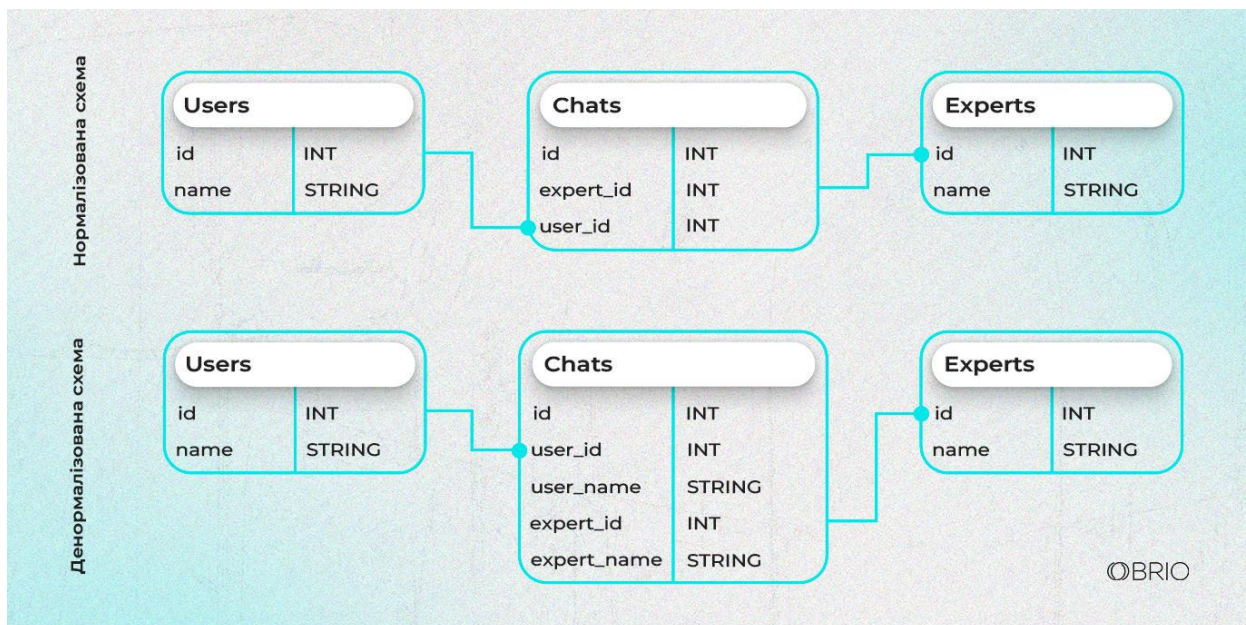


Рис. 2.5. Діаграма денормалізації

Цей метод – досить ефективний, але потрібно відчувати межу, щоби дублі не повторювалися занадто часто. Інакше це призведе до неконсистентності даних.

Переваги:

швидкість запити: підвищення продуктивності вибірки за допомогою уникнення низки операцій JOIN в запиті;

спрощена структура бази даних, якщо правильно застосувати цей метод.

Недоліки:

дублювання даних;

збільшення об'єму даних на диску: через дублювання обсяг зберігання може зростати, що підвищує вартість інфраструктури;

неконсистентність даних: при дублюванні існує ризик, що оновлення може не пройти всюди, де треба;

складність: забагато дублікатів може ускладнити систему, тому цей метод не можна використовувати надмірно.

Метод горизонтального партиціювання передбачає розділення таблиці на сегменти (партиції) на одному сервері за визначеною логікою, наприклад за часом створення запису. Зазвичай у такому підході ви звертаєтесь до однієї віртуальної таблиці, а СУБД вирішує, до якої фізичної таблиці звертатися.

Наприклад, велика база даних для вебмагазину зберігає історію покупок користувачів. Таблиця Замовлення містить мільйони рядків, її обробка та аналіз із часом сповільнюються та стають неефективними. Крім того, резервне копіювання та відновлення такої великої таблиці може бути часозатратним. Можна розділити її на партиції за місяцями або роками. Наприклад, одна частина буде містити всі замовлення за січень 2023 року, інша – за лютий 2023 року тощо. Запити до БД, які обробляють лише певний діапазон даних (наприклад, замовлення за певний місяць), будуть швидше виконуватися, оскільки вони звертатимуться лише до однієї партиції, а не до усїєї великої таблиці. Операції з управління даними (наприклад, видалення старих замовлень) можуть бути спрощені, оскільки можна просто видалити цілу частину.

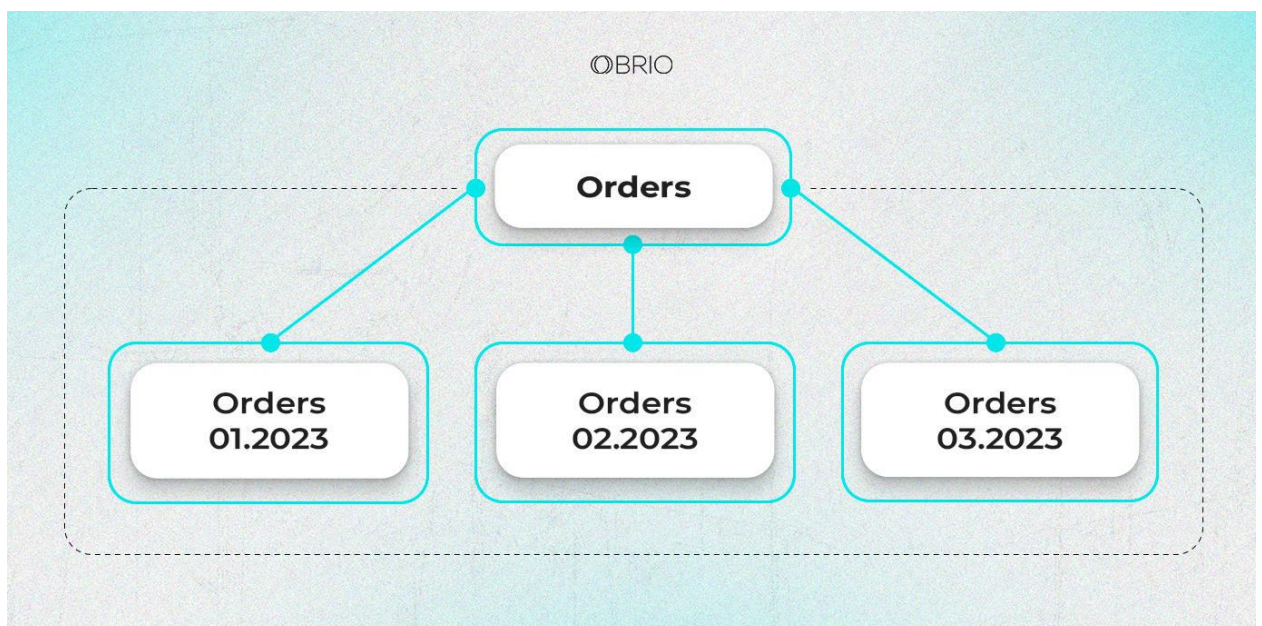


Рис. 2.6. Діаграма метод горизонтального партиціювання

Переваги:

зменшення розміру індексів, що суттєво прискорює операцію запису;
швидке видалення даних за певною ознакою, що не блокує інші частини таблиці;

простота налаштування та використання, адже зазвичай СУБД дає це «з коробки».

Недоліки:

складність керування транзакціями: коли дані розподілені між підтаблицями, це може призвести до додаткових труднощів, особливо при роботі з декількома партиціями одночасно;

цей підхід не впливає на доступність системи: якщо сервер БД впаде, вся система не буде працювати.

Отже, кожен підхід має свої переваги та недоліки, і вибір залежить від конкретних вимог та цілей бізнесу. Адаптація та гнучкість у виборі методу масштабування дозволяють командам оптимізувати витрати ресурсів, забезпечити стабільність та ефективність роботи їхніх систем.

2.2. Методичні підходи до оптимізації баз даних

Квази-оптимальний план виконання декларативного запиту – це найефективніший спосіб виконання запиту в контексті можливостей Системи управління базами даних (СУБД).

Процес оптимізації запитів в реляційних СУБД включає два різних аспекти. По-перше, це внутрішня задача СУБД, яка полягає в визначенні найбільш оптимального (ефективного) методу виконання реляційних запитів. По-друге, це завдання програміста (або кваліфікованого користувача): написання реляційних запитів таким чином, щоб СУБД могла використовувати більш ефективні методи отримання даних. Розглянемо перший аспект.

Етапи оптимізації запитів в реляційних СУБД

Мова SQL є декларативною мовою. Її командах відсутня інформація про те, як виконати запит, які методи доступу до даних використовувати. Майже кожна команда SQL із підмножини мови маніпулювання даними (DML) може бути виконана різними способами. Розглянемо наступний запит:

Метою СУБД є виконання запиту, причому зробити це якнайефективніше. Таким чином, під оптимізацією розуміється побудова квазі-оптимального процедурного плану виконання декларативного запиту.

Примітка: квазіоптимальний план вважається таким, оскільки система не гарантує, що вона для будь-якого запиту вибере оптимальний план. Для гарантованого вибору оптимального плану необхідно розглянути всі можливі варіанти планів і порівняти їх, а це може зайняти більше часу, ніж виконання самого запиту. Таким чином, СУБД вибирає і аналізує лише кілька планів виконання кожного запиту, і серед них може не виявитися оптимального плану.

План виконання запиту складається з послідовності кроків, кожен з яких або фізично витягує дані з пам'яті, або виконує підготовчу роботу. Побудовою цього плану займається оптимізатор – спеціальний компонент СУБД. Обробка запиту, який надійшов в РСУБД і представлений на декларативній мові запитів, складається з етапів (фаз).

На першій фазі запит, викладений мовою запитів, піддається лексичному та синтаксичному аналізу. Лексичний аналізатор розбиває запит на лексичні одиниці – лексеми (найменування полів та таблиць, константи, знаки операцій тощо). Синтаксичний аналізатор перевіряє синтаксичну правильність запиту. У результаті формується внутрішнє представлення запиту. Воно відображає структуру запиту і містить інформацію, яка характеризує об'єкти бази даних, згадані в запиті (таблиці, поля, константи). Інформація про об'єкти бази даних вибирається зі словника-довідника даних. Внутрішнє представлення запиту використовується і трансформується на наступних етапах обробки запиту.

На другій фазі запит в своєму внутрішньому представленні піддається логічній оптимізації. При цьому можуть застосовуватися різноманітні перетворення, «вдосконалюючи» початкове представлення запиту. Серед цих перетворень можуть бути еквівалентні перетворення. Після проведення еквівалентних перетворень отримується внутрішнє представлення, семантично еквівалентне початковому запиту.

В будь-якому випадку після виконання другої фази обробки запиту його внутрішнє представлення залишається непроцедурним, хоча і в певному сенсі ефективнішим, ніж початкове. Це означає, що за цим представленням система зможе побудувати більш ефективний план.

Третя фаза обробки запиту полягає у виборі альтернативних процедурних планів виконання даного запиту відповідно до його внутрішнього представлення, отриманого на другій фазі. Оптимізатор обирає методи доступу, послідовність з'єднань та методи з'єднань для генерації ефективного плану виконання запиту. Інформація про наявні шляхи доступу до даних береться зі словника-довідника даних. Єдиний можливий шлях доступу, який можливий у будь-якому випадку, – це послідовне читання (FULL). Можливість використання інших шляхів доступу залежить від способів розміщення даних в пам'яті (наприклад, кластеризації або хешування даних), від наявності індексів та формулювання самого запиту.

Також на третьому етапі для кожного з обраних планів оцінюється передбачувана вартість виконання запиту за цим планом. При оцінках використовується або доступна оптимізатору статистична інформація про розподіл даних, або інформація про механізми реалізації шляхів доступу. З альтернативних планів вибирається найбільш оптимальний з точки зору якогось (заздалегідь вибраного або заданого) критерію.

На четвертому етапі для внутрішнього представлення найбільш оптимального плану виконання запиту формується процедурне представлення плану. Виконуване представлення плану може бути програмою в машинних кодах, якщо, як у випадку System R, система

орієнтована на компіляцію запитів в машинні коди. У інших системах, наприклад, в INGRES, представлення плану є машинно-незалежним, що є більш зручним для інтерпретації запитів. Для нас це не принципово, оскільки четвертий етап обробки запиту вже не пов'язаний з оптимізацією.

Нарешті, на останньому, п'ятому етапі обробки запиту відбувається його реальне виконання відповідно до процедурного плану виконання запиту. Результат зберігається в спеціальній області оперативної пам'яті (так званому курсорі).

Тепер розглянемо трансформації, яким може бути піддано вихідний реляційний запит на етапі логічної оптимізації.

Трансформації операцій реляційної алгебри. Операндами операцій реляційної алгебри (РА) є відношення, тобто неупорядковані множини кортежів. Для виконання операцій необхідно проглянути всі кортежі вихідного відношення (або відношень). Внаслідок цього операції РА мають великі розміри. Зменшення розмірів можна досягти, змінюючи послідовність виконуваних операцій. В якості прикладу розглянемо відношення $R1$ і $R2$, які містять по 1000 кортежів, при цьому лише 10 кортежів в кожному відношенні задовольняють умові F . Якщо виконувати наступну послідовність операцій:

$\sigma F (R1 \cup R2)$, то після виконання об'єднання отримаємо 2000 кортежів (якщо відношення не містять однакових кортежів), а після селекції залишиться 20 записів. Якщо змінити послідовність виконання операцій:

$\sigma F (R1) \cup \sigma F (R2)$, то після селекції залишиться по 10 записів з кожного відношення, об'єднання яких дасть 20 потрібних кортежів. Якщо врахувати, що об'єднання зазвичай виконується шляхом сортування даних (для вилучення однакових кортежів) і проміжний результат потрібно зберігати, то виграш як по обсягу пам'яті, так і за часом очевидний: набагато швидше відсортувати 20 кортежів, ніж 2000.

Трансформація операцій РА базується на понятті еквівалентності реляційних виразів. Операндами виразів є змінні-відношення R_i та

константи. Кожний вираз реляційної алгебри визначає відображення кортежів змінних-відношень R_i ($i=1, \dots, n$) в кортежі єдиного відношення, яке отримується в результаті підстановки кортежів кожного R_i та виконання всіх визначених виразом обчислень.

Два вирази реляційної алгебри вважаються еквівалентними, якщо вони описують одне й те саме відображення. Існують закони, які, відповідно до цього визначення, дозволяють виконувати еквівалентні перетворення виразів реляційної алгебри:

Закон комутативності для декартових добутків:

$$R1 \times R2 = R2 \times R1.$$

Закон комутативності для з'єднань (F – умова поєднання):

$$R1 \bowtie_F R2 = R2 \bowtie_F R1.$$

Закон асоціативності для декартових добутків:

$$(R1 \times R2) \times R3 = R1 \times (R2 \times R3).$$

Закон асоціативності для поєднань ($F1, F2$ – умови поєднання):

$$(R1 \bowtie_{F1} R2) \bowtie_{F2} R3 = R1 \bowtie_{F1} (R2 \bowtie_{F2} R3).$$

Комбінація селекцій (каскад селекцій):

$$\sigma_{F1} (\sigma_{F2} (R)) = \sigma_{F1 \wedge F2} (R).$$

Комбінація проєкцій (каскад проєкцій):

$$\pi_{A1, A2, \dots, Am} (\pi_{B1, B2, \dots, Bn} (R)) = \pi_{A1, A2, \dots, Am} (R), \text{ де } \{A_m\} \subset \{B_n\}.$$

Перестановка селекції і проєкції:

$$\sigma_F (\pi_{A1, A2, \dots, Am} (R)) = \pi_{A1, A2, \dots, Am} (\sigma_F (R)).$$

Перестановка селекції с поєднанням:

$$\sigma_F (R1 \cup R2) = \sigma_F (R1) \cup \sigma_F (R2).$$

Перестановка селекції с декартовим добутком:

$$\sigma_F (R1 \times R2) = (\sigma_{F1} (R1)) \times (\sigma_{F2} (R2)),$$

(якщо $F = F1 \wedge F2$, где $F1$ має в собі атрибути, наявні тільки в $R1$, а $F2$ має в собі атрибути, наявні тільки в $R2$);

$$\sigma_F (R1 \times R2) = (\sigma_F (R1)) \times R2,$$

(якщо F має в собі атрибути, наявні тільки в R1);

$$\sigma_F (R1 \times R2) = R1 \times (\sigma_F (R2)),$$

(якщо F має в собі атрибути, наявні тільки в R2);

$$\sigma_F (R1 \times R2) = \sigma_{F2} (\sigma_{F1} (R1) \times R2),$$

(якщо $F = F1 \wedge F2$, где F1 мають в собі атрибут, наявні тільки в R1, а F2 має в собі атрибути, наявні і в R1, и в R2). Перестановка селекції з різницею:

$$\sigma_F (R1 - R2) = \sigma_F (R1) - \sigma_F (R2).$$

Перестановка селекції з перетином: $\sigma_F (R1 \cap R2) = \sigma_F (R1) \cap \sigma_F (R2)$.

2.3. Кібербезпека даних як складова оптимізації реляційних баз даних

Швидка цифровізація впливає на всі аспекти життя – включаючи спосіб взаємодії, роботу, покупки та отримання послуг – а також на те, як створюється та змінюється ціна. У цьому процесі дані та транскордонний потік даних стають все більш важливими для розвитку. Хоча традиційний цифровий розрив відображає значні відмінності у готовності використовувати дані між країнами та всередині них, він пов'язаний із комунікаціями. Цей розрив поглиблюється так званим обміном даними. Країни з обмеженою спроможністю перетворювати дані в цифровий інтелект і можливості для бізнесу та використовувати їх для економічного та соціального розвитку явно знаходяться в невідповідному становищі.

Однією з основних проблем цифровізації є забезпечення захищеності даних. Захист даних – це набір процесів і практик, призначених для захисту вашої екосистеми критичних інформаційних технологій. Це включає файли, бази даних, облікові записи та мережі. Ефективна безпека даних передбачає набір елементів керування, програм та методів, які визначають важливість різних наборів даних та застосовують найбільш відповідні засоби безпеки.

Вона враховує чутливість різних наборів даних та відповідні нормативні вимоги піддатливості.

Як і інші позиції з кібербезпеки (приклад: безпека периметру та файлів) – безпека даних не є кінцевою для запобігання хакерським атакам. Отже, безпека даних – один із багатьох критичних методів оцінки загроз та зменшення ризику, пов'язаного із зберіганням та обробкою даних. Аналіз останніх досліджень і публікацій.

Питанню кібербезпеки бізнесу та держав приділяють увагу багато наукових діячів. Так, Безуглий Д.С. обґрунтував необхідність інформаційної безпеки як складової частини національної безпеки країни. У вітчизняній юридичній літературі дослідженню окремих питань цієї проблематики в різні часи приділяли увагу такі фахівці, як А. Марущак, В. Брижко, М. Різак, В. Панченко, О. Радкевич та інші. Аналіз останніх досліджень і публікацій свідчить про те, що певні аспекти вітчизняних проблем інформаційної безпеки у той чи інший спосіб досліджувались у наукових працях Арістова І.В., Березовської І.Р., Дзьобаня О.П., Калюжного Р.А., Кормича Б.А., Ліпкана В.А., Марущак А.І., Цимбалюка В.С., Юдіна О.К. та інших. Питанням формування ефективного механізму правового регулювання протидії загрозам у кібернетичній сфері присвятили свої праці такі науковці: Бурячок В.Л., Шеломенцев В.П., Сопілко І.В., Куцаєв В.В., Мінін Д.С., Гнатюк С.О. та інші. Проте ці дослідження здебільшого зосереджені на сфері правового регулювання та формування системи інформаційної безпеки України. Розгляд цього питання здійснюється і зарубіжними вченими – І. Вельдер, А. Міллер, Р. Холлборг та інші (Dr. Christopher Karvetski, Necati Demir PhD, Oliver Holloway, IBM, John Tsitsiklis (professor MIT), Dimitri Bertsekas (professor MIT), Patric Jaillet (professor MIT)). Виділення невіршених раніше частин загальної проблеми.

Сьогодні, в контексті глобальної цифровізації, питання захисту є надзвичайно важливими і порушуються в більшості країн. Навіть інформаційні гіганти не завжди можуть повністю контролювати чи

підтримувати узгодженість та конфіденційність своїх даних/програмного забезпечення, що свідчить про важливість цієї теми. Не менш важливу проблему становить сьогодні торгівля базами даних, мова йде про інформацію як комерційного, так і особистого характеру. У багатьох країнах, незважаючи на різні підходи до цього питання, частина відомостей закрита для публічного розповсюдження, проте досить часто вона продається особами, які мають до неї доступ. Формування цілей статті (постановка завдання). Організації по всьому світу вкладають значні кошти в інформаційні технології кібербезпеки, щоб захистити свої критично важливі активи. Незалежно від того, чи потрібно компанії захищати свій бренд, інтелектуальний капітал та інформацію про клієнтів, чи контролювати критичну інфраструктуру, інструменти виявлення інцидентів та пропаганди мають три загальні елементи: люди, процеси та технології.

З різних причин безпека даних має вирішальне значення для державного та приватного секторів. По-перше, компанія має юридичні та моральні зобов'язання захищати дані своїх користувачів і клієнтів від потрапляння в чужі руки. Наприклад, фінансовій компанії може знадобитися дотримуватися стандарту безпеки даних індустрії платіжних карток (PCI DSS), який змушує компанію вживати всіх розумних заходів для захисту даних користувачів. По-друге, існує ризик шахрайства з репутацією або шахрайством з даними. Якщо компанія не ставиться до безпеки даних серйозно, її репутація може бути назавжди зіпсована рекламою, гучними витоками чи хакерством. Не кажучи вже про фінансові та матеріально-технічні наслідки вторгнення даних. Компанії повинні витратити час і гроші на оцінку та усунення втрат, а також на визначення того, які бізнес-процеси виявилися невдалими, а які необхідно покращити.

Цифрова економіка – це система економічних та соціальних відносин, які формуються на навичках інформаційно-комп'ютерних технологій (ІКТ) для ефективного виробництва, продажу та постачання продуктів та здійснення ділових операцій на ринку. Вона продовжує розвиватися з

великою швидкістю, керованою здатністю збирати, використовувати та аналізувати велику кількість інформації. Цифрові дані виникають із потоків особистої, соціальної та ділової діяльності, що відбувається на різних цифрових платформах. Окрім перспектив за рахунок цифровізації, виникає питання надійності в збереженні усіх даних та ступінь їх захищеності. Дані стають активом. Збирання, опис, зберігання та опрацювання їх надають змогу отримувати цінну інформацію для використання в ділових процесах, суспільному житті, роботі держави. Вміння працювати з великими даними та їх аналізувати – це можливість першим отримувати цінні ринкові «інсайти», тобто бути конкурентоздатнішим. Доступ до даних здійснюється через Інтернет та інші мережі. Велика частина даних у світі стає відкритою. Проаналізувавши цю таблицю, необхідно знайти розумний і прийнятний баланс між особистою участю в інформаційному просторі та повагою до особистої свободи та прав громадян в Інтернеті. Оскільки все більше комп'ютерів підключаються до Інтернету, у публічний інформаційний простір потрапляє все більше особистої інформації. Цифрові дані фактично не регулюються, оскільки вони не підпадають під жодні правила, які застосовуються до звичайних товарів.

Імовірність неправомірного використання інформації дуже висока і збільшується через адміністративні рішення або комерційні операції. Необхідно вживати заходів, що дозволяють, якщо не повністю виключити, то хоча б мінімізувати цю небезпеку. Багато приватних компаній давно займаються питаннями безпеки власної продукції в цифровому просторі. У січні 2002 року голова правління Microsoft Білл Гейтс звернувся до 50 тис. співробітників корпорації з листом, в якому поставив завдання створити для клієнтів захищене інформаційне середовище, яке було б вкрай надійне. Програма захищених інформаційних систем охоплює чотири напрямки: безпека, конфіденційність, безвідмовність і бізнес-етику.

На державному рівні також вживаються заходи, направлені на захист цифрового простору. В Україні діє Закон «Про захист інформації в

інформаційно-телекомукаційних системах» від 05.07.1994 № 80/94-ВР. Останні зміни до Закону були прийняті 4 червня 2020 року, які були напрацьовані робочою групою Державної служби спеціального зв'язку та захисту інформації України, Міністерством цифрової трансформації України, Державною митною службою та експертів галузі. Ці зміни стосуються підтвердження відповідності інформаційної системи вимогам із захисту інформації ЄС. Впроваджено вимоги стандартів сімейства системи управління інформаційною безпекою (СУІБ) для окремих категорій інформації. Ці вимоги діятимуть для забезпечення інформаційної та кібербезпеки в ЄС і наблизять Україну до європейських норм. За Законом, підтвердження відповідності комплексної системи захисту інформації (КСЗІ) відбуватиметься за результатами держекспертизи. Відтепер державні інформаційні ресурси та інформація з обмеженим доступом, крім державної таємниці, може оброблятися в системі без застосування КСЗІ, у разі виконання умов:

- підтвердження відповідності системи управління інформаційною безпекою національним стандартам України щодо систем управління інформаційною безпекою;

- використання для захисту інформації в системі засобів криптографічного захисту інформації, які мають позитивний експертний висновок за результатами державної експертизи;

- жоден з елементів системи не може бути розташований на території України, де органи державної влади тимчасово не здійснюють своїх повноважень, на територіях держав, визнаних Верховною Радою України державами-агресорами, на територіях держав, щодо яких застосовані санкції відповідно до Закону України «Про санкції», та на територіях держав, які належать до митних союзів з такими державами;

- виконання особливих вимог, встановлених Урядом до забезпечення захисту інформації в системах залежно від категорії державних інформаційних ресурсів або інформації з обмеженим доступом, вимога щодо

захисту якої встановлена законом. На сьогодні Інтернет-платформа є потужною, але вона не призначена для справжньої децентралізації. Більшість сучасних обчислень та зберігання в Інтернеті виконується у хмарних сховищах та центрах обробки даних великих корпорацій. Їх продуктивність і надійність виявляються набагато сильнішими, ніж особисті термінали. В результаті більшість прав та цінностей користувачів Інтернету захоплюються цими корпораціями.

Бар'єри для розвитку тренду в Україні:

– Відсутність системи правил, регламентів, стандартів збирання, класифікації, зберігання та використання даних (національний, регіональний, галузевий та інші рівні).

– Проблеми захисту інтелектуальної власності.

– Проблеми щодо захисту даних, ризику кібербезпеки.

– Відсутність у громадян достатніх компетентностей роботи з даними (цифрових навичок), відповідної освіти, професій тощо. Можливості, які створює тренд для України

– Розвиток нової галузі економіки, нові робочі місця.

– Створення бази для розвитку всіх галузей та цифрової економіки.

– Поява ефективного інструменту управління.

– Створення середовища, що унеможливорює корупцію як явище.

Завдяки послугам на вимогу та масштабованості хмарні обчислення стали новою технологією. Сьогодні вони найчастіше використовуються для зберігання даних і великих програм або обчислювальних програм. Тому безпека та конфіденційність даних стали основними проблемами, особливо для даних бізнес-рівня. Враховуючи глобальний характер світової економіки, слід зазначити, що питання цифрової безпеки стає наднаціональним. Типовими загрозами національній цифровій безпеці є фінанси, енергетика та інфраструктура.

Потенційна уразливість цифрових систем створює небезпеку цифрового колапсу, подібного до того, який описали в своїй книзі Люсі та

Стівен Хокінг. При цьому можливе загострення соціальних проблем: зростання структурного безробіття як наслідок цифрової революції, відчуження працівника від продукту своєї праці в результаті розвитку дистанційної зайнятості, особисте збіднення індивідів. Інноваційні та інформаційні ресурси мають позанаціональний характер, застосовуються в різних країнах, тому захист даних – проблема не тільки національного, а й наднаціонального рівня. У хмарному середовищі дані користувачів зберігаються у віддалених екземплярах віртуальних машин, що знаходяться у розпорядженні хмарних провайдерів. Можуть бути різні зовнішні атаки на віртуальні машини, включаючи атаку шкідливих кодів, компрометацію відповідного монітора віртуальної машини тощо. Крім зовнішньої атаки, користувачі не мають фізичного контролю над своїми даними.

Інсайдери хмарних провайдерів могли чітко бачити, що зберігається у їхніх екземплярах віртуальних машин. Було б катастрофою, якщо інсайдери хмарних провайдерів змовилися з супротивниками, щоб навмисно змінити або випустити дані клієнтів. Існують три основні елементи безпеки даних, яких повинні дотримуватися всі організації: конфіденційність, цілісність та доступність.

Ці концепції також називаються Тріадою ЦРУ, яка функціонує як модель безпеки та рамка для першокласної безпеки даних. Ось що означає кожен основний елемент з точки зору захисту ваших конфіденційних даних від несанкціонованого доступу та вилучення даних. Інструменти та технології захисту даних мають вирішувати зростаючі проблеми, властиві безпеці сучасних складних, розподілених, гібридних та/або багатохмарних обчислювальних середовищ. Вони включають розуміння, де знаходяться дані, відстеження того, хто має до них доступ, а також блокування діяльності з високим ризиком та потенційно небезпечних переміщень файлів.

Комплексні рішення щодо захисту даних, які дозволяють підприємствам прийняти централізований підхід до моніторингу та застосування політики, можуть спростити завдання. Конфіденційна

інформація може міститися у структурованих та неструктурованих хмарних сховищах. Рішення для виявлення та класифікації даних автоматизують процес виявлення конфіденційної інформації, а також оцінки та усунення вразливих місць. Інструменти виявлення та класифікації даних:

– Моніторинг активності даних та файлів. Інструменти моніторингу активності аналізують шаблони використання даних, дозволяючи групам безпеки бачити, хто та для чого звертається до даних, виявляти аномалії та визначати ризики.

– Інструменти оцінки вразливості та аналізу ризиків. Ці рішення полегшують процес виявлення та пом'якшення таких вразливих місць, як застаріле програмне забезпечення, неправильна конфігурація або слабкі паролі, а також можуть визначати джерела даних з найбільшим ризиком зараження.

– Автоматизоване звітування про відповідність вимогам. Комплексні рішення для захисту даних з можливостями автоматизованої звітності можуть створити централізоване сховище для перевірок відповідності на рівні підприємства. Дослідження Інституту Понемона щодо вартості даних та щодо порушення даних виявило, що в середньому збиток, завданий внаслідок порушення даних у США, становив 8 мільйонів доларів. 25575 облікових записів користувачів зазнали впливу середнього інциденту з даними, що означає, що крім фінансових втрат більшість інцидентів призводять до втрати довіри клієнтів та шкоди репутації. Великий відсоток «викриття» даних не є результатом зловмисної атаки, а спричинене необережним або випадковим викриттям конфіденційних даних.

Співробітники організації часто розповсюджують, надають доступ, втрачають або неправильно поведуться з цінними даними – випадково або тому, що вони не знають про політику безпеки. Цю велику проблему можна вирішити шляхом навчання співробітників, а також за допомогою інших заходів, таких як технологія запобігання втраті даних (DLP) та покращений контроль доступу. Середня вартість порушень даних є найвищою в США.

Яскравим та актуальним прикладом проблеми захисту даних є тимчасове відключення основного сайту соціальних мереж Facebook Inc, популярної платформи обміну фотографіями Instagram та додатку для обміну повідомленнями WhatsApp, які були заблоковані десяткам тисяч користувачів. Reuters, одне з найбільших у світі міжнародних агентств новин та фінансової інформації, не змогло підтвердити проблему, яка впливала на платформу. Однак повідомлення про помилку на веб-сторінці Facebook передбачає помилку системи доменних імен (DNS). DNS дозволяє веб-адресам доставляти користувачів до місця призначення.

Подібний збій у хмарній компанії Akamai Technologies Inc знищив кілька веб-сайтів у липні. Перебої в технологіях це не рідкість, але водночас затьмарити стільки програм від найбільшої у світі компанії з соціальних медіа було надзвичайно незвично. Останнє значне відключення Facebook відбулося в 2019 році, коли технічна помилка вплинула на його сайти протягом 24 годин. Цього разу винуватцем стали зміни базової інфраструктури Інтернету, яка координує трафік між її центрами обробки даних. Це призвело до переривання зв'язку та каскаду до інших центрів обробки даних, зупинивши сайт. Через цей інцидент акції Facebook впали на 5,5% у другій половині дня в понеділок, наблизившись до свого найгіршого показника за майже рік.

Захист даних – це захід безпеки, призначений для запобігання несанкціонованому доступу до баз даних, веб-сайтів і комп'ютерів. Цей процес також забезпечує механізм захисту даних від втрати або пошкодження. Кожен бізнес сьогодні, чи то маленька компанія, місцева компанія чи велика компанія, не може забути вжити заходів безпеки. Відсутність належного плану інформаційної безпеки може призвести до серйозних наслідків для бізнесу. Це важливо з основних причин: забезпечити безперервність бізнесу, уникнути витоку даних та запобігти несанкціонованому доступу.

Сьогодні важливість безпеки даних зростає в кожній галузі. Безпека даних у особистому житті потрібна кожному. Кожна компанія має у своїй базі даних великі дані, які необхідно захищати. Якщо цю інформацію будь-яким чином знищити, компанія зазнає величезних збитків. Можна сказати, що «дані – це нова нафта». У будь-якому бізнесі найголовніше – це його дані. Тому, щоб захистити ці дані, компанії також готові використовувати високий рівень безпеки з високою ціною. Використовуючи ці методи для запобігання витоку даних, ви можете мінімізувати ризик і захистити себе від будь-яких кібератак. Тенденції безпеки даних:

- Штучний інтелект покращує можливості систем безпеки даних, оскільки він може обробляти великі обсяги даних. Когнітивні обчислення – це підмножина штучного інтелекту, який виконує ті ж завдання, що й інші системи штучного інтелекту, але досягається шляхом моделювання процесу мислення людини. У сфері безпеки даних це дозволяє швидко приймати рішення, коли це потрібно терміново.

- Холодна безпека. Зі збільшенням хмарних функцій розширюється і визначення безпеки даних. Зараз організаціям потрібні складніші рішення, оскільки їм потрібно не лише захищати дані, а й додатки та власні бізнес-процеси, які працюють у публічних та приватних хмарах.

- Квантова. Очікується, що революційна технологія Quantum змінить багато традиційних технологій у геометричній прогресії. Алгоритми шифрування стануть різноманітнішими, складнішими та безпечнішими.

Цінність бізнесу та захист даних ніколи не були вищими, ніж сьогодні. Втрата комерційної таємниці або інтелектуальної власності (ІВ) може вплинути на майбутні інновації та прибутковість. Крім того, надійність стає все більш важливою для споживачів: захист персональних даних і впевненість у надійності їх діяльності в Інтернеті стають пріоритетними. Це особливо важливо в умовах нинішньої швидкої оцифровки всіх аспектів життя. Додавання додаткового рівня безпеки даних

може значно запобігти несанкціонованому доступу до компанії або обладнання та систем.

Коли немає фізичних загроз, то й фізичний захист не потрібний, а ось інформаційний – потрібен завжди. Що таке інформаційна безпека бізнесу і що потрібно робити, щоб звести нанівець ризики як внутрішніх, так і зовнішніх атак? Чому в сучасному світі ми все частіше звертаємо увагу на цей вид безпеки? Розповіла Mind CEO компанії SOFTICO Лариса Куроєдова.

Інформаційна безпека бізнесу стала найважливішою складовою його успішного ведення. І починається вона вже лише з однієї думки створити компанію. Під час розвитку бізнес постійно змінюється, йому доводиться підлаштовуватися під реалії часу, зважати на нові правила гри і дотримуватися букви закону. Керівництво компаній часто двояко ставиться до безпеки бізнесу: хтось із більшою відповідальністю, а хтось скептично. Іноді загрози переоцінюють, іноді вони недооцінюються. Великий бізнес піддається атакам десятки разів на день, причому ці атаки мутують. Але зараз ми говоримо не про захист від зовнішніх атак, а про створення безпечного інформаційного середовища в самому бізнесі.

Фахівці в галузі інформаційної безпеки (ІБ) краще за всіх розуміють, що захист інформаційної інфраструктури організації відкриває нові ділові можливості, а наявні бізнес-процеси вимагають менше ресурсів для ефективної роботи. Надійний захист інформації дозволяє залучити до бізнесу нових партнерів. Чим вищий рівень довіри, тим більший рівень доступу безпечно можна надавати зовнішнім сторонам: клієнтам, діловим партнерам, співробітникам і підрядникам. Це допомагає розширити бізнес та одночасно спрощує виконання операцій, знижуючи витрати.

Однак класичний підрозділ з ІБ, на жаль, дуже часто відірваний від реалій бізнесу та прив'язаний суто до технологій. Компанії, у яких директори та керівники відділів з ІБ мають уявлення про бізнес-процеси та методологію управління проектами, можна полічити на пальцях. Крім того, чомусь існує стереотип, що ІБ прив'язана до інформаційних технологій (ІТ). Добре, що в

пресі та на семінарах все частіше почали наголошувати, що ІБ – це не лише ІТ. Але знову ж таки, згідно зі статистикою, у більшості компаній ІБ прив'язана до ІТ, має загальний з ІТ бюджет, а топменеджмент компаній упевнений, що ІБ – лише підрозділ ІТ.

Необхідно розуміти, що в сучасному світі ІБ – це насамперед дуже сильний менеджмент, чітке уявлення бізнес-стратегії компанії та вміння взаємодіяти з усіма її підрозділами. Технології – це лише інструмент. Без ефективного менеджменту й розуміння бізнес-стратегії компанії на підрозділи ІБ дивитимуться як на підрозділи, що не впливають на фінансові результати компанії.

Управління ІБ на середньому рівні дозволяє окупити витрати, а за хорошого менеджменту – збільшувати прибутки та робити бізнес ще ефективнішим. Сьогодні йдеться про те, що ІБ дозволяє розширювати бізнес у глобальному масштабі незалежно від розмірів компанії або її розташування.

Саме тому до питання збереження даних будь-якого бізнесу варто ставитися надзвичайно серйозно.

ІТ-експерт, співзасновник та член Наглядової ради міжнародного об'єднання Integracy Group Антон Марреро розповів, що обидві ці сфери взаємодіють та взаємопов'язані і є ключовими компонентами успішного функціонування сучасних бізнесів. Розглянемо, чому ці аспекти є настільки важливими для комерційних підприємств.

1. **Захист конфіденційної інформації:** Комерційні підприємства зберігають значну кількість конфіденційних даних, таких як особиста інформація клієнтів, фінансові дані, бізнес-секрети та інші важливі ресурси. Втрата чи розголошення цих даних може призвести до фінансових втрат, репутаційної шкоди та юридичних проблем. Захист цих даних є важливим завданням, яке можна досягти завдяки належній інформаційній безпеці та кібербезпеці.

2. **Забезпечення надійності операцій:** Комерційні підприємства відчужуються від традиційних бізнес-моделей та переходять до цифрових систем і мереж. Будь-яке порушення роботи цих систем може призвести до перерв у роботі, що супроводжується втратами прибутку та погіршенням обслуговування клієнтів. Кібербезпека допомагає запобігти таким перервам, забезпечуючи надійність операцій.

3. **Захист від кіберзагроз:** Сучасні кіберзлочинці постійно вдосконалюють свої методи атак та шукають нові цілі для вторгнень. Бізнеси повинні бути готові до різних загроз, таких як віруси, шкідливі програми, фішинг, атаки на мережу та інші. Кібербезпека надає захист від цих загроз та допомагає відновлювати системи в разі атаки.

4. **Дотримання законодавства:** Багато країн встановлюють обов'язкові вимоги щодо захисту конфіденційної інформації та звітності про кіберінциденти. Комерційні підприємства повинні дотримуватися цих правил, а недотримання може призвести до штрафів та інших юридичних наслідків.

5. **Збереження репутації:** Репутація є однією з найцінніших активів для будь-якого комерційного підприємства. Втрата довіри клієнтів може негативно вплинути на бізнес та його прибуток. Інформаційна та кібербезпека допомагає уникнути витоку даних та інцидентів, які можуть вразити репутацію.

Висновки до другого розділу

Отже, розділ, присвячений аспектам забезпечення інформаційної та кібербезпеки в комерційних підприємствах, висвітлив ряд ключових викликів, які стикаються сучасні бізнеси у цифровому світі. Захист конфіденційної інформації, забезпечення надійності операцій, захист від кіберзагроз, дотримання законодавства та збереження репутації визначаються як критичні аспекти управління ризиками та забезпечення стабільності бізнес-процесів.

Ключовою тезою є важливість впровадження комплексних стратегій забезпечення інформаційної безпеки, які охоплюють технологічні, організаційні та правові аспекти. Втрата конфіденційної інформації може призвести до серйозних наслідків, таких як фінансові втрати, порушення законодавства та пошкодження репутації.

Водночас, визначено, що кібербезпека визначається не лише як профілактичний захід, але і як важлива складова відновлення підприємства після кібератак. Сучасні загрози вимагають постійного удосконалення заходів безпеки та готовності до реагування на нові виклики.

Зазначено, що дотримання законодавства у сфері кібербезпеки є необхідною умовою для уникнення штрафів та інших правових наслідків. Зокрема, це стосується вимог щодо захисту конфіденційної інформації та звітності про кіберінциденти.

Нарешті, зазначено, що репутація є ключовим активом для комерційних підприємств, і заходи з інформаційної та кібербезпеки визначаються як ефективні інструменти для уникнення витоку даних та збереження довіри клієнтів. Розділ підкреслив, що інвестування в ці аспекти є стратегічно важливим для успішної діяльності в умовах сучасного бізнес-середовища.

РОЗДІЛ 3

ПРАКТИЧНА РЕАЛІЗАЦІЯ СИСТЕМИ ОПТИМІЗАЦІЇ БАЗИ ДАНИХ В БІЗНЕСІ

3.1. Створення автоматизованої інформаційної системи підприємства

Мета даної роботи: створення автоматизованої інформаційної системи для обліку діяльності малого фітнес-клубу, яка дозволить легко вести списки клієнтів та співробітників, складати супутні документи, а також отримувати звіти, що характеризують стан справ в організації на певний період.

Для різноманітних підприємств, від маленьких фірм до найбільших мереж, є багато готових програмних продуктів.

Всі ці продукти можна абстрактно розділити на групи:

- корпоративні інформаційні системи (КІС), системи планування ресурсів підприємства (ERP -системи), системи обліку;
- інформаційно-аналітичні системи (ІАС), системи для розроблення планів, моделей;
- системи управління взаємовідносинами з клієнтами (CRM системи).

Системи планування ресурсів підприємства та корпоративні інформаційні системи не підходять, що називається за визначенням, для малих підприємств.

Так для впровадження ERP -системи знадобляться не тільки великі фінансові витрати, але і велика кількість трудових і тимчасових ресурсів, оскільки такі системи мають складний інтерфейс користувача. Та й взагалі, системи планування ресурсів мають низьку інтеграційну здатність, що є вкрай важливим для АІС малого бізнесу. КІС не можуть бути використані практично з тих самих причин, проте вони інтегровані. Простіше кажучи, дані системи не враховують галузеву специфіку та масштаби ведення бізнесу.

Інформаційно-аналітичні системи, в порівнянні з ERP-системами та КІС, простіше для впровадження і, як правило, набагато дешевше. На ринку такі системи представлені у вигляді коробкових і напівкоробкових рішень. ІАС та системи для розробки моделей часто входять до управлінських програмних комплексів, але такі програмні продукти не по кишені малому бізнесу.

У цілому нині, малий бізнес має низку особливостей, які впливають працювати з інформаційними технологіями. Так, вузько спрямована спеціалізація фітнес-клубів створює складність застосування розгалуженої інформаційної структури, а специфіка галузі породжує додаткові фінансові витрати на її розробку.

Сектор малих підприємств має проблему обмеженості ресурсів, що дозволяє їм впроваджувати дорогі програмні продукти, які б створити зручну інформаційну інфраструктуру. Якщо фінансова можливість і є, то придбане серйозне програмне рішення, розраховане на великі фірми, буде далеко не найкращим вкладенням. На малих підприємствах широкий функціонал реалізації бізнес-процесів буде використаний не повністю, велика кількість функцій буде зайве і зовсім не застосовується.

Крім того, більшість готових програмних засобів складні для розуміння та роботи навіть для впевненого користувача комп'ютера. Така ситуація спричинить додаткові витрати на навчання персоналу.

Через те, що мале підприємство, в даному випадку фітнес-клуб, має обмеженість у робочих кадрах, при автоматизації необхідно наголосити на полегшенні здійснення його діяльності, а саме, необхідно розробити інтегроване програмне забезпечення. Продукт, розроблений за принципом "Все в одному", дозволить об'єднати безліч різних функцій [3].

Програмний продукт, який може бути впроваджений на малому підприємстві, має відповідати низці вимог, найважливіші серед яких:

- бюджетна вартість продукту та подальшого його супроводу;
- простота впровадження та експлуатації;

- інтеграція коїться з іншими системами;
- відсутність функцій, у яких немає потреби.

Наприклад, всі описані вище характеристики поєднують у собі CRM - системи. Вони дозволять не втрачати клієнтів, і чітко знати, що потрібно кожному їх.

Система управління взаємовідносинами з покупцями призначена для оптимізації процесів взаємодії з клієнтами на всіх стадіях роботи. Найчастіше CRM-системи застосовують у організаціях, які працюють за принципами прямих продажів, тобто. продаж послуг безпосередньо клієнту.

Впровадивши CRM -систему, співробітники різних відділів, наприклад, адміністратори фітнес-клубу, отримують програмний продукт із зручним інтерфейсом для виконання своїх обов'язків по роботі з клієнтами, а директор, у свою чергу – можливість здійснення контролю за підлеглими та звітністю.

Сучасний ринок пропонує кілька варіантів CRM-систем для малих фірм:

- програмний продукт, встановлений на персональний комп'ютер;
- серверна система;
- Хмарна система - найбільш вдале рішення.

Виходить, що спектр комп'ютерних програм для великих та середніх компаній досить широкий, що не можна сказати про підприємства малого бізнесу. Незважаючи на, здавалося б, велику кількість готових програмних засобів, невеликим організаціям вигідніше й практичніше розробляти власне програмне забезпечення, яке відповідатиме необхідним у кожному конкретному випадку вимогам і не матиме незатребуваного функціоналу.

На даний момент існує безліч програмних рішень у сфері управління бізнесом, включаючи продукти для ведення обліку на підприємствах типу фітнес-клуб. Порівняємо деякі з них.

Розробка програмного забезпечення під конкретну фірму є найбільш вигідним варіантом для підприємств малого бізнесу, ніж встановлення та обслуговування готового рішення з надлишковим функціоналом.

Для проектування автоматизованої системи обліку діяльності фітнес клубу було обрано діаграми нотації ArgoUML, а також CASE засоби ERwin Data Modeler та Ramus Educational .

Пакет ArgoUML – рішення для моделювання UML діаграм, що вільно розповсюджується. Пакет пропонує інтуїтивно зрозумілий графічний інтерфейс, що не вимагає тривалого вивчення. Пропонований набір діаграм, що перекриваються, дозволяє отримати повне уявлення про проект. Пакет ArgoUML не пропонує всієї різноманітності функцій більш розгорнути пакети, такі як System Architect компанії Popkin Software та Paradigm Plus компанії Computer Associates . Однак має унікальні можливості: фоновий аналіз під час роботи, а також випуск пакету у вихідних текстах.

Для моделювання даних предметної області було використано CASE - засіб Ramus Educational Ramus - це програма, за допомогою якої можна створювати візуальні діаграми, що використовуються для наочного відображення різних бізнес-процесів. Програма містить безліч інструментів, які покращують зовнішній вигляд проектів, просте графічне середовище зі зрозумілою оболонкою, можливість створення звітів та довідкових документів в автоматичному режимі.

Головною перевагою Ramus Educational полягає в тому, що вона підтримує відразу дві популярні методології: DFD та IDEF0, а також можливість редактора графічних файлів із зручною панеллю інструментів та великою базою шаблонів.

Для проектування бази даних веб-застосунку було використано CASE засіб ERwin Data Modeler .

ERwin – CASE - засіб для проектування та документування баз даних, що дозволяє створювати, документувати та супроводжувати бази даних, сховища та вітрини даних. Моделі даних допомагають візуалізувати

структуру даних, забезпечуючи ефективний процес організації, управління та адміністрування таких аспектів діяльності підприємства, як рівень складності даних, технологій баз даних та середовища розгортання. ERwin поєднує графічний інтерфейс Windows, інструменти для побудови ER-діаграм, редактори для створення логічного та фізичного опису моделі даних та прозору підтримку провідних реляційних СУБД та настільних баз даних. За допомогою ERwin можна створювати чи проводити зворотне проектування баз даних.

До переваг пакету ERwin Data Modeler можна віднести такі можливості:

- взаємодія користувачів через веб-портал;
- стандартизовані уявлення імен об'єктів, стандартизовані типи даних та стандартні зразки моделей (референтні моделі);
- графічне представлення моделей даних різної структури та ступеня вкладеності за рахунок потужної графічної системи та системи навігації;
- автоматичний контроль версій, керування доступом, керування конфліктами та зміною моделей.

Автоматизуємо різноманітні види діяльності, використовуючи єдину технологічну платформу. В основних рисах варто виділити такі особливості:

- за допомогою цієї платформи можна оптимізувати фірми з будь-якою кількістю робочих місць;
- платформа може функціонувати в операційних системах Windows, Linux, FreeBSD ;

Важливим етапом проектування є побудова моделі бізнес – процесів. На початкових етапах створення ІВ необхідно зрозуміти, як працює організація, яку необхідно автоматизувати. Тому для опису роботи підприємства необхідно побудувати модель, яка відповідатиме предметній галузі та міститиме у собі знання всіх учасників бізнес-процесів організації.

Контекстна діаграма бізнес-процесів інформаційної системи «Діяльність фітнес-клубу» у методології IDEF0 представлена рис. 3.1.

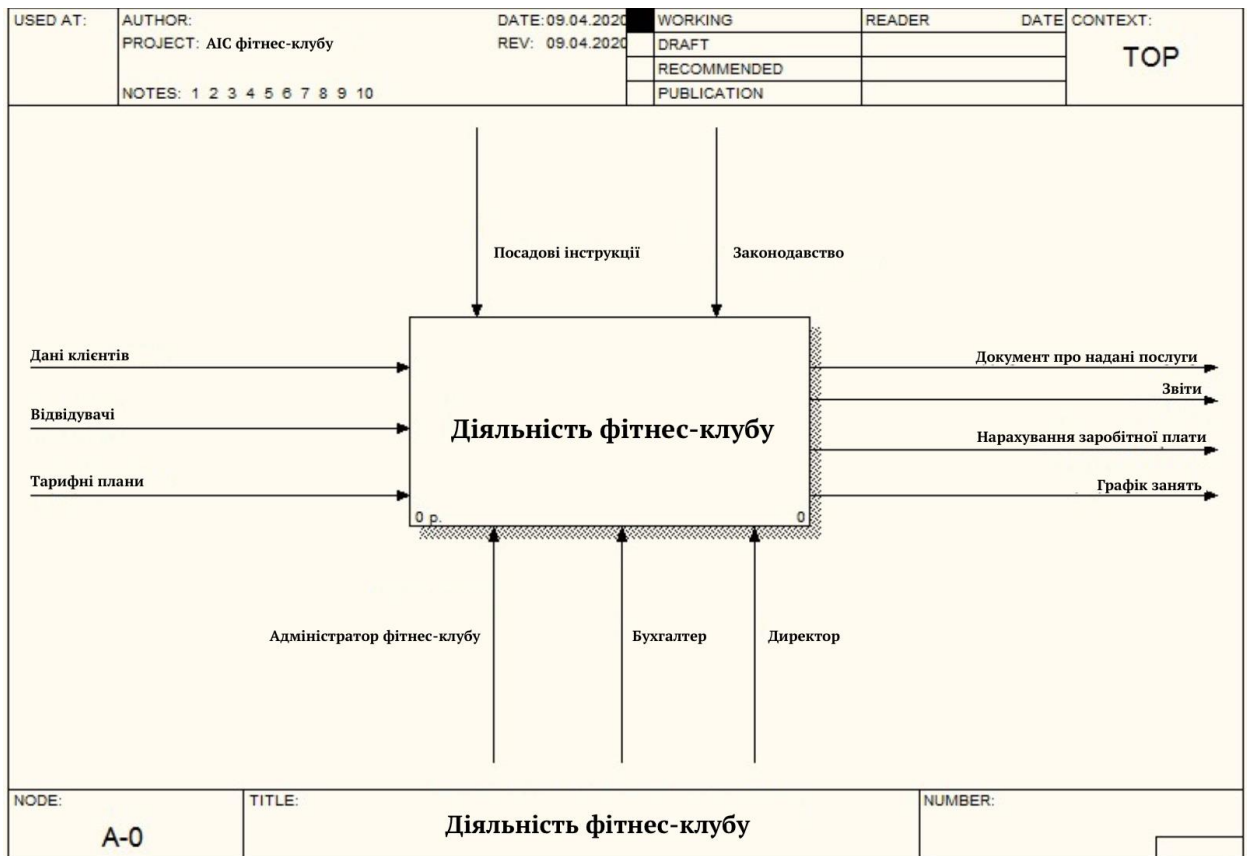


Рис. 3.1. Контексна діаграма

У цій контекстній діаграмі вхідними даними є:

- дані клієнта;
- відвідувачі;
- тарифні плани.

Вихідними даними для бізнес-процесу «Діяльність фітнес-клубу» є:

- документ про надані послуги;
- звіти;
- нарахування заробітної плати;
- графік занять.

Механізми та управління це чинне законодавство, посадові інструкції та інструкції з охорони праці та техніки безпеки. Ресурсами є: персонал, ПК та офісна техніка, інформаційне забезпечення та сервісне обладнання.

На основі створеної контекстної діаграми в результаті деталізації узагальненого процесу "Діяльність фітнес-клубу" була розроблена деталізуюча діаграма, представлена на рисунку 3.2.

У процесі декомпозиції бізнес-процесу системи «Діяльність фітнес-клубу» виділено 6 процесів: «Облік клієнтів», «Продаж абонементів», «Відвідування фітнес-клубу», «Облік заробітної плати працівникам», «Складання графіка занять», «Формування звітів».

Процес «Облік клієнтів» призначений для фіксації особистих даних про клієнтів з можливістю фільтрації даних за різними показниками. Також зберігати повну хронологію роботи з кожним клієнтом та в два кліки отримувати доступ до даних будь-якого покупця.

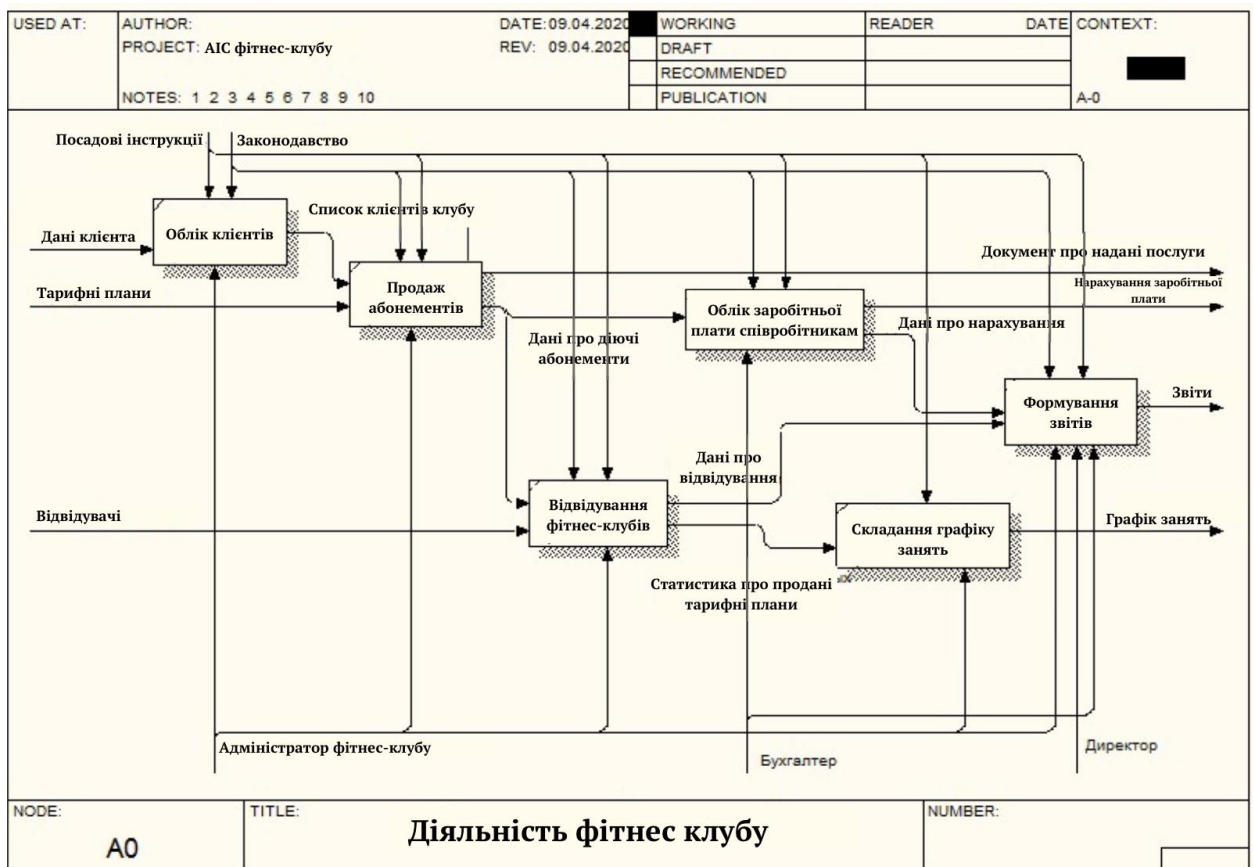


Рис. 3.2 Діалізуюча діаграма процесу «Діяльність фітнес клубу» у нотації IDEF0

Процес «Продаж абонементів» призначений для обліку продажів абонементів клієнтам. На даному етапі після звернення клієнта менеджер з продажу проводить консультацію щодо чинних тарифних планів. Після цього клієнт вирішує сплатити чи відмовитися від покупки. Менеджер з продажу

обдзвонює клієнтів з пропозиціями про діючі тарифи та знижки. Після узгодження вартості та виду тарифної плати, клієнт укладає договір і оплачує його вартість. З оплати інформація заноситься в БД і клієнту видається іменна карта відвідувача.

Процес «Відвідування фітнес-клубу» призначений для обліку відвідувань фітнес-клубу. Цей процес передбачає облік разових тренувань, списання тренувань з чинних акаунтів членів клубу та пакетів послуг, статистика відвідувань послуг.

Процес «Облік заробітної плати співробітникам» призначений для обліку нарахувань регламентованої заробітної плати працівникам та розрахунку встановлених законодавством податків та внесків із фонду оплати праці.

Процес «Упорядкування графіка занять» призначений для складання розкладу групових тренувань, а також можливість доповнювати графік індивідуальними тренуваннями.

Процес «Формування звітів» призначений для складання та формування різних видів звітів, включаючи звіти щодо клієнтів, фінансових результатів, роботи співробітників.

Оскільки всі процеси діаграми декомпозиції бізнес-процесу «Діяльність фітнес-клубу» є простими, їх подальша декомпозиція не потрібна.

Логічна модель даних є початковим зразком майбутньої бази даних. Вона будується у термінах інформаційних одиниць, але не прив'язки до конкретної СУБД. Логічна модель описує поняття предметної області, їх взаємозв'язок, і навіть обмеження дані, накладені предметної областю [10]. У процесі розроблення логічної моделі даних виділено 14 сутностей.

Логічна модель даних представлена рис. 3.3.

Було розроблено логічну модель даних для інформаційної системи фітнес-центру. Ця логічна модель містить 14 сутностей.

Було визначено реквізити сутності та первинні ключі.

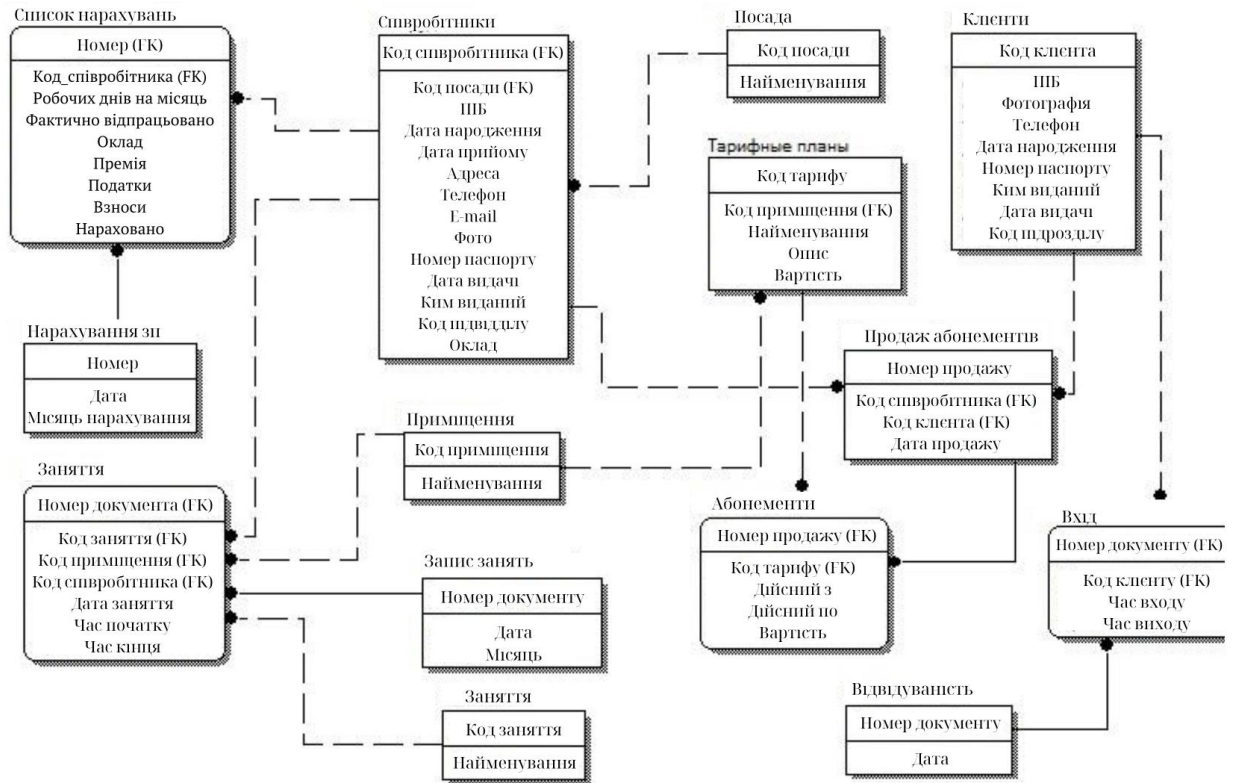


Рис. 3.3. Логічна модель даних

При проектуванні таблиць та визначенні зв'язків між ними використовують спосіб нормалізації. Цей спосіб дозволяє розділити вихідну, суцільну таблицю ряд елементарних таблиць, між якими встановлюються зв'язку. Така сукупність пов'язаних таблиць створює єдиний інформаційний ланцюг об'єкта. Разом з тим з цього ланцюга можна вибирати часткову інформацію про об'єкт, що зберігається в окремій таблиці. У той самий час зв'язку дозволяють здійснити і зворотний процес «складання» інформації з інформації однієї з таблиць [11].

Опис структури зв'язків логічної моделі бази даних представлено табл. 3.1. Таким чином, були визначені зв'язки між сутностями та їх типом даних представлених зв'язків.

Опис структури зв'язків

Головна сутність	Залежна сутність	Назва зв'язку	Потужність
Клієнти	Продаж абонементів	Неідентифікуюча	Один до багатьох
Клієнти	Вхід	Неідентифікуюча	Один до багатьох
Відвідуваність	Вхід	Ідентифікуюча	
Продаж абонементів	Співробітники	Неідентифікуюча	
Продаж абонементів	Абонементи	Ідентифікуюча	
Співробітники	Посада	Неідентифікуюча	Один до багатьох
Співробітники	Список нарахувань	Неідентифікуюча	Один до багатьох
Співробітники	Заняття	Неідентифікуюча	Один до багатьох
Нарахування з/п	Список нарахувань	Ідентифікуюча	
Розклад занять	Список занять	Неідентифікуюча	
Список занять	Приміщення	Неідентифікуюча	
Список занять	Заняття	Неідентифікуюча	Один до багатьох
Тарифні плани	Абонементи	Неідентифікуюча	Один до багатьох
Тарифні плани	Приміщення	Неідентифікуюча	Один до багатьох

Для підтримки БД у стійкому стані використовують ряд механізмів, які отримали узагальнену назву засобів підтримки цілісності. Процес нормалізації і є приведення структури БД у відповідність до цих обмежень.

В цілому суть цих обмежень дуже проста: кожен факт, що зберігається в БД, повинен зберігатися один раз, оскільки дублювання може призвести до неузгодженості між копіями однієї і тієї ж інформації. Слід уникати будь-яких неоднозначностей, а також надмірності інформації, що зберігається.

Нормалізацією схеми бази даних називається процедура, вироблена над базою даних із метою видалення надмірності.

Виділяються шість нормальних форм, п'ять з яких так і називаються: перша, друга, третя, четверта, п'ята нормальна форма, а також нормальна форма Бойса -Кодда, що лежить між третьою та четвертою [12].

Для реляційної моделі даних розроблено кілька нормалізованих форм, три з яких є основними.

База даних вважається нормалізованою, якщо її таблиці представлені як мінімум у третій нормальній формі. Часто багато таблиць нормалізуються до четвертої нормальної форми, іноді, навпаки, проводиться денормалізація. Використання таблиць у п'ятій нормальній формі у реальних базах даних трапляється рідко [13].

Перша нормальна форма (1НФ) говорить, що кожен атрибут відносини повинен зберігати атомарне значення, кожне відношення (рядок у таблиці) має містити однакову кількість атрибутів (стовпців), тобто:

- забороняє повторювані стовпці (що містять однакову за змістом інформацію);
- забороняє множинні стовпці (що містять значення типу списку тощо);
- вимагає визначити первинний ключ для таблиці, тобто той стовпець чи комбінацію стовпців, які однозначно визначають кожен рядок.

Друга нормальна форма (2НФ) говорить, що відношення знаходиться в другій нормальній формі, якщо воно знаходиться в 1НФ, і при цьому всі ключові атрибути залежать тільки від первинного ключа, тобто:

- друга нормальна форма вимагає, щоб ключові стовпці таблиць залежали від первинного ключа загалом, але з його частини;
- якщо таблиця перебуває у першій нормальній формі і первинний ключ у неї складається з одного стовпця, вона автоматично перебуває й у другій нормальній формі.

Відношення знаходиться у третій нормальній формі (3НФ), якщо воно знаходиться у другій нормальній формі, і кожен не ключовий атрибут залежить тільки від первинного ключа і не залежать один від одного.

Розроблена логічна модель даних інформаційної системи фітнес-клубу знаходиться у третій нормальній формі. Це наступним.

1. Усі сутності логічної моделі даних системи перебувають у 1НФ, оскільки всі атрибути атомарні і з даних відносин має первинний ключ.

Розглянемо деякі сутності для підтвердження цього факту:

– сутність «Приміщення» має ключовий атрибут «Код приміщення» та атрибут «Найменування», які є простими та неподільними;

– сутність «Розклад занять» має ключовий атрибут «Номер документа» та атрибути «Дата», «Місяць», які є простими та неподільними;

– сутність «Тарифні плани» має ключовий атрибут «Код тарифу» та атрибути «Код приміщення», «Найменування», «Опис», «Вартість», які є простими та неподільними;

– сутність "Клієнти" має ключовий атрибут "Код клієнта" та атрибути "ПІБ", "Фотографія", "Телефон", "Дата народження", "Номер паспорта", "Ким виданий", "Дата видачі", "Код підрозділу", які є простими та неподільними;

– сутність «Нарахування з/п» має ключовий атрибут «Номер» та атрибути «Дата», «Місяць нарахування», які є простими та неподільними.

Інші сутності також перебувають у першій нормальній формі.

2. 2НФ вимагає, щоб неключові атрибути відносин залежали від первинного ключа загалом, але з його частини. Розглянемо деякі сутності на підтвердження цього факту.

Сутність "Список занять" має складовий первинний ключ, що складається з атрибутів "Номер документа", "Код заняття". Неключові атрибути цього відношення залежать від первинного ключа в цілому, а не від його частини. Атрибути "Дата заняття", "Час початку", "Час закінчення" "Код приміщення", "Код співробітника" залежать і від Номер документа, і від Код заняття. Отже, можна дійти невтішного висновку, що це сутність перебуває в 2НФ.

Сутність "Абонементи" має складовий первинний ключ, що складається з атрибутів "Номер продажу", "Код тарифу". Неключові

атрибути цього відношення залежать від первинного ключа в цілому, а не від його частини. Атрибут «Дійсний», «Дійсний», «Вартість» залежить і від Номер продажу, і від Код тарифу. Отже, можна зробити висновок, що дана сутність перебуває в 2НФ.

Сутність "Список нарахувань" має складовий первинний ключ, що складається з атрибутів "Номер", "Код співробітника". Неключові атрибути цього відношення залежать від первинного ключа в цілому, а не від його частини. Атрибут "Робочих днів на місяці", "Фактично відпрацьовано", "Оклад", "Премія", "Податки", "Внески", "Нараховано" залежать від Номера та від Код співробітника. Отже, можна дійти невтішного висновку, що це сутність перебуває в 2НФ.

Інші сутності також знаходяться у другій нормальній формі.

3. У жодному з відносин логічної моделі немає транзитивних залежностей, тобто. неключові атрибути не залежать функціонально друг від друга, тому відносини перебувають у 3НФ.

Розглянемо деякі сутності на підтвердження цього факту.

Щодо «Співробітники» неключові атрибути «ПІБ», «Дата народження», «Дата прийому», «Адреса», «Телефон», «Email», «Фотографія», «Номер паспорта», «Дата видачі», «Ким виданий», «Код підрозділи», «Оклад» функціонально не залежить один від одного.

Що стосується «Тарифні плани» неключові атрибути «Код приміщення», «Найменування», «Опис», «Вартість» функціонально залежить від друг друга.

Що стосується «Вхід» неключові атрибути «Код клієнта», «Час входу», «Час виходу» функціонально залежить від друг друга.

Щодо «Клієнти» неключові атрибути «Код клієнта» та атрибути «ПІБ», «Фотографія», «Телефон», «Дата народження», «Номер паспорта», «Ким виданий», «Дата видачі», «Код підрозділу» функціонально не залежать один від одного.

Що стосується «Продаж абонементів» неключові атрибути «Код співробітника», «Код клієнта», «Дата продажу» функціонально залежить від друг друга.

Таким чином, можна зробити висновок, що розроблена модель знаходиться у третій нормальній формі.

3.2. Проектування програмного забезпечення системи обліку діяльності фітнес-клубу

У процес проектування ПЗ входять розробка моделей, діаграм, що відбивають структуру програми, їх текстовий опис. Все це складає проектну документацію. Для її створення може бути використана мова UML – уніфікована мова моделювання, призначена для опису програмних систем.

Недоліком мови є початкова орієнтованість на об'єктно-орієнтовані системи, що накладає обмеження на виконання основної своєї функції – надати, з одного боку, досить формальний, з іншого боку, досить зручний, і, з третього боку, досить універсальний засіб, що дозволяє до певної міри знизити ризик розбіжностей у тлумаченні специфікацій [14].

Проектування системи здійснюється з використанням UMLдіаграм за допомогою CASE -засобу ArgoUML .

Загальна функціональність системи UML представляється з допомогою діаграми прецедентів. Вона дозволяє показати користувачів системи та її функції. Основними елементами діаграми є актори та варіанти використання (прецеденти), з якими вони взаємодіють. Між прецедентами можуть існувати стосунки. Діаграма прецедентів представлена малюнку 4.

Опис дійових осіб представлено табл.3.2, а короткий опис варіантів використання табл. 3.3.

Список дійових осіб

Найменування особи	Профіль, підготовка та навички
Адміністратор	Регулярний користувач інформаційної системи, особа з великим досвідом роботи з подібним програмним забезпеченням, займається роботою з клієнтами: заводить особисті картки, продає абонементи (при цьому оформляється документ, що містить дані про придбання), формує рахунки на оплату для клієнта та відстежує їхню оплату.
Бухгалтер	Регулярний користувач інформаційної системи, особа з великим досвідом роботи з подібним програмним забезпеченням, ведеться картотека співробітників та оформлюються документи щодо нарахування заробітної плати.
Керівник	Особа, що має досвід роботи з даними програми, ознайомена з вузьким функціоналом програми, займається розробкою прайс-листа та цінової політики, а також проводить аналіз звітів діяльності фірми за результатами роботи.

Опис варіантів використання

Прецедент	Суб'єкти	Дія
Авторизація	Директор	Користувач здійснює вхід до системи
	Адміністратор	
	Бухгалтер	
Перегляд даних	Директор	Директор переглядає дані щодо клієнтів (їх відвідуваність, продані абонементи) та співробітників (оклади співробітників, нараховані з/п)
	Адміністратор	Адміністратор може переглянути прайс-лист та відвідуваність клієнтів
	Бухгалтер	Бухгалтер має право переглядати прайс-лист
Ведення довідкових даних	Директор	Директор веде прайс-лист клубу та встановлює на них ціни
Звіт з прайсу	Директор	Користувачі формують звіт із актуальним прайс-листом, запровадивши період; є можливість друку
	Адміністратор	

Звіт з відвідуваності	Директор	Користувачі формують звіт за кількістю відвідувачів, ввівши період, та вибирають тип подання (у вигляді таблиці/у вигляді діаграми); є можливість друку
	Адміністратор	
Звіт з послуг	Директор	Користувачі формують звіт із рейтингом послуг з виручки, ввівши період, і вибирають тип подання (як таблиці/ як діаграми); є можливість друку
	Адміністратор	
	Бухгалтер	
Звіт про продовження абонементів	Адміністратор	Адміністратор клубу формують звіт зі списком клієнтів, у яких добігає кінця термін дії абонементу, ввівши період, і вибирають тип подання (у вигляді таблиці/ у вигляді діаграми); є можливість друку
Робота з клієнтами	Адміністратор	Адміністратор веде картотеку клієнту та продає абонементи.
Робота зі співробітниками	Бухгалтер	Бухгалтер веде картотеку співробітників, встановлює оклади та нараховує з/п

У табл. 3.4 наведено відомості про функції учасників бізнес-процесу і документи, що передаються.

Таблиця 3.4

Відомості про функції учасників бізнес-процесу та документи, що передаються

Учасники бізнес-процесу	Функції компонента	Одержувач даних	Документи
Адміністратори	Надання інформації про вартість послуг Видача документа зі списком отриманих послуг Формування звітів щодо рейтингу послуг та відвідуваності	Клієнт	Прайс лист Продаж абонементів
		Керівництво	Аналіз відвідуваності Рейтинг послуг Продаж абонементів
		Бухгалтерія	Рейтинг послуг
Бухгалтерія	Нарахування заробітної плати працівникам Установка окладу персоналу	Керівництво	Нарахування заробітної плати Оклад співробітників

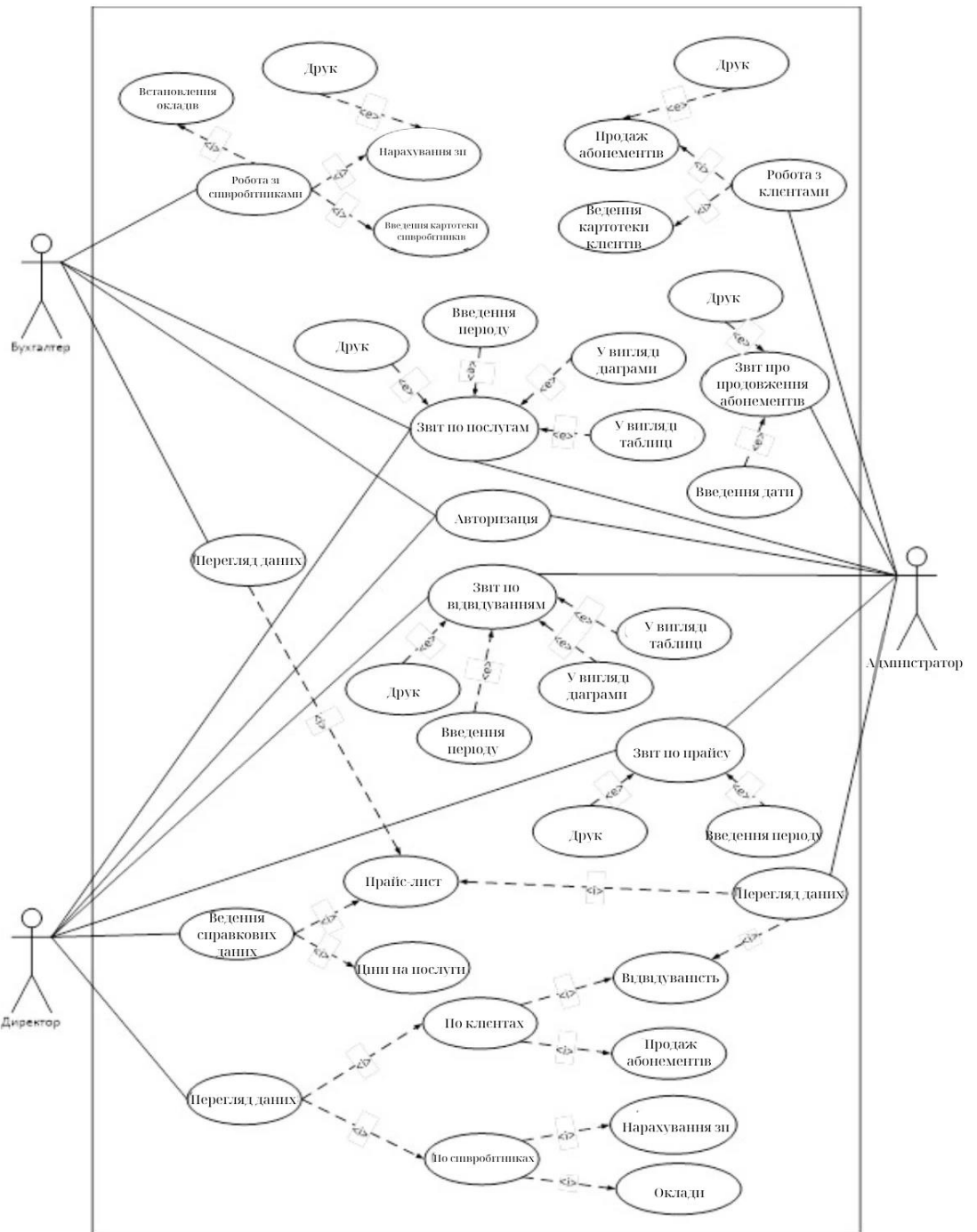


Рис. 3.4. Діаграма варіантів використання

В результаті аналізу були виявлені та описані прецеденти системи, а також побудована діаграма варіантів використання.

Стан – це з характеристик об'єктів системи, поруч із поведінкою. Кожен об'єкт є набір атрибутів, виконуваних функцій і має взаємозв'язку коїться з іншими об'єктами [15]. Стан визначається значеннями перерахованих вище

параметрів у певний момент роботи системи. Зміна стану відбувається при настанні будь-якої події (наприклад, після надсилання повідомлення). Діаграма станів показує, як об'єкт переходить з одного стану до іншого і яка подія при цьому відбулася

У цій системі діаграма детально відображає зміну стану документа "Продаж абонементів". Діаграма стану в нотатції UML представлена рис. 3.5.

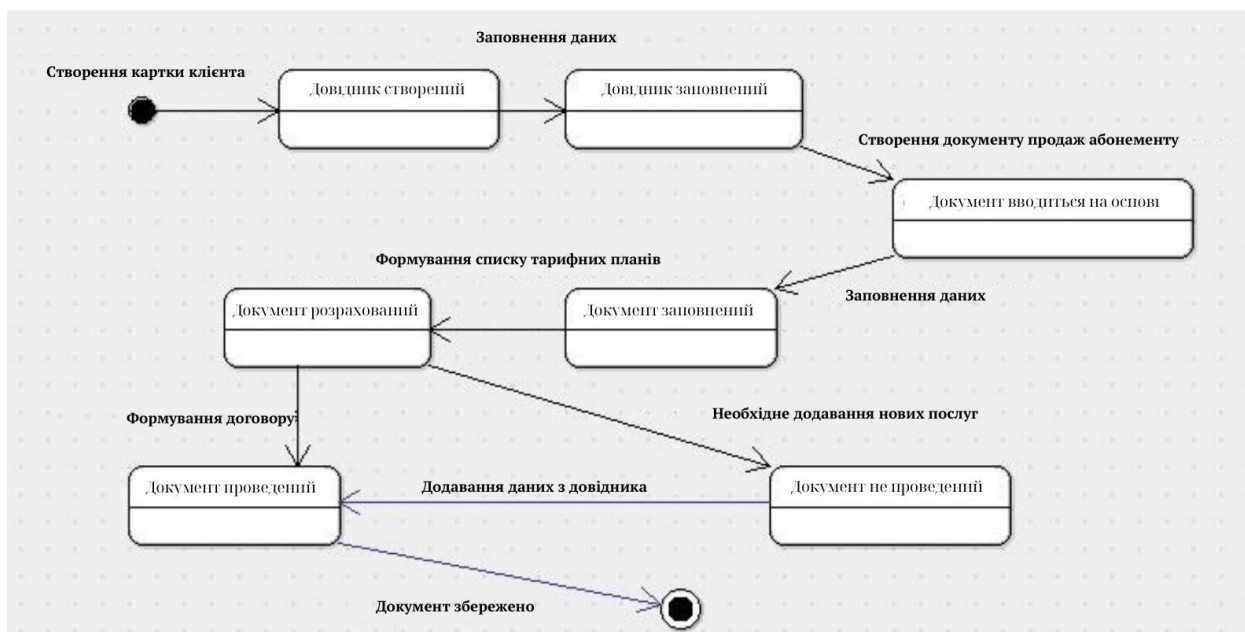


Рис. 3.5. Діаграма стану

Виділено такі стани об'єкта: довідник створено, довідник заповнено, документ вводиться на підставі, документ створено, заповнено дані документа, розраховано тарифні плани, документ проведено, документ не проведено.

Діаграма діяльності – UML -діаграма, на якій показано розбиття деякої діяльності на її складові. Під діяльністю розуміється специфікація виконуваної поведінки у вигляді координованого послідовного та паралельного виконання підлеглих елементів – вкладених видів діяльності та окремих дій, з'єднаних між собою потоками, що йдуть від виходів одного вузла до входів іншого. Діаграми діяльності використовуються при

моделюванні бізнес-процесів, технологічних процесів, послідовних та паралельних обчислень [16].

У цій системі діаграма детально відображає процес придбання абонементу. Діаграма діяльності у нотації UML представлена рис. 3.6.

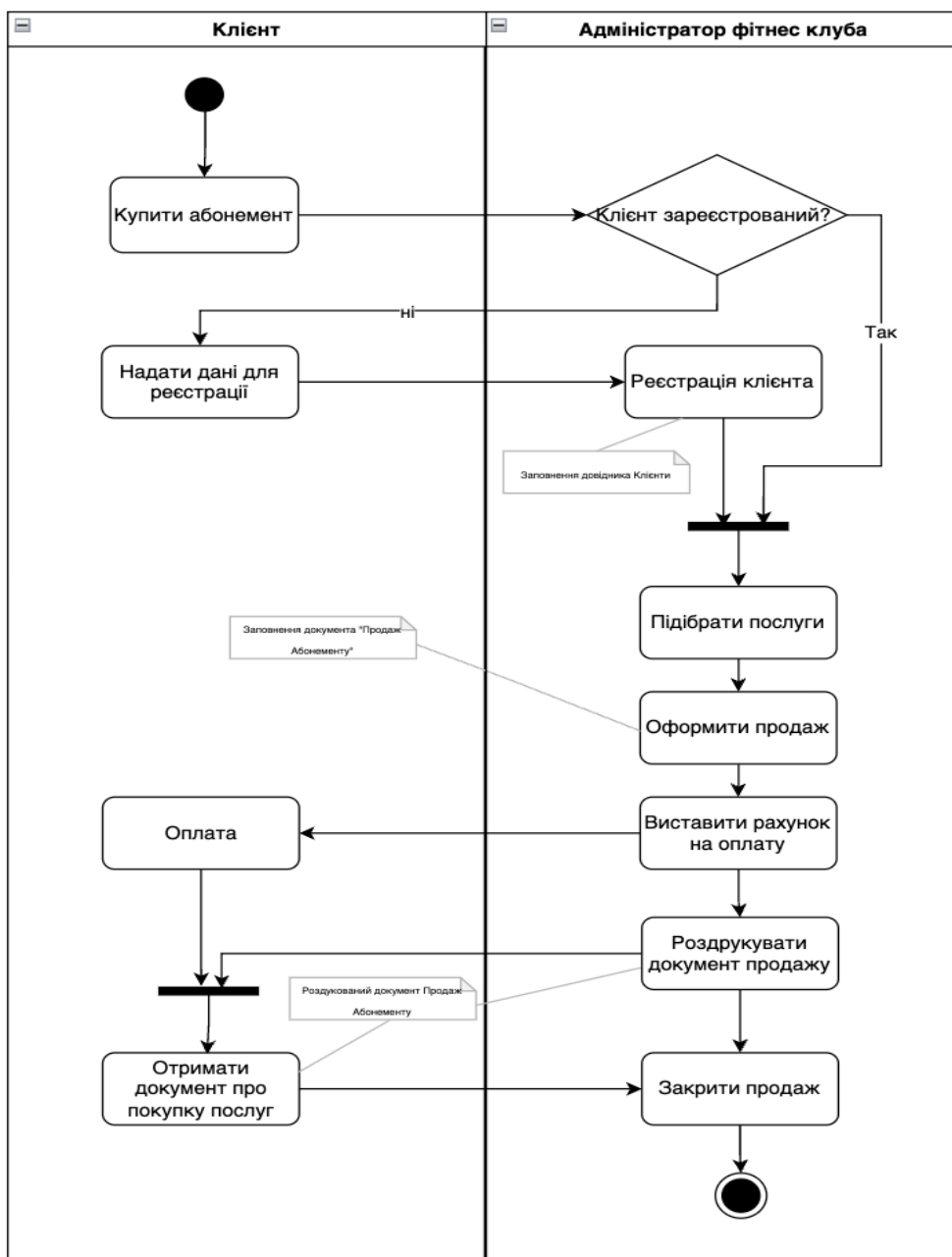


Рис. 3.6. Діаграма діяльності

Необхідно адміністратору зареєструвати клієнта в інформаційну базу, далі підібрати тарифний план та оформити продаж абонементу. За підсумками продажу формується друкований договір та акт виконаних робіт.

Діаграма послідовності описує сценарій роботи програми. На діаграмі відображаються екземпляри об'єктів та повідомлення, якими вони обмінюються в рамках одного варіанта використання.

У цій системі виділимо 3 об'єкти («Менеджер», документ «Продаж абонементів» та реєстр накопичення «Продаж»). Діаграма відображає процес створення документа про продаж абонементів клієнтам та запис відповідних даних до реєстру. Діаграма послідовності у нотатції UML представлена рис. 3.7.

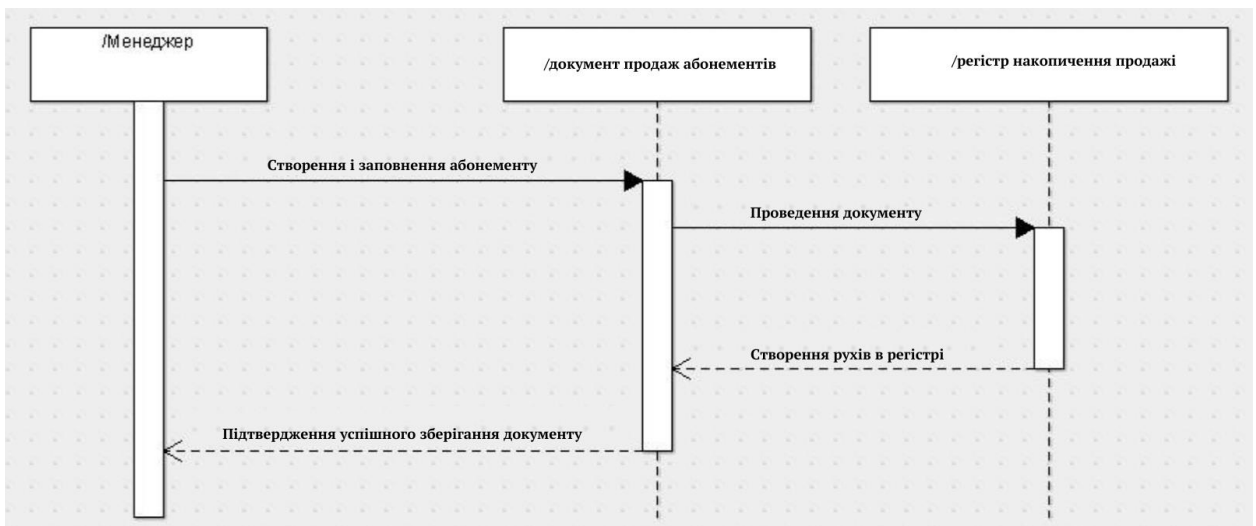


Рис. 3.7. Діаграма послідовності

Усі об'єкти, які представлені на діаграмі, існують протягом усього сеансу спілкування.

Поняття кооперації (collaboration) є одним із фундаментальних понять у мові UML. Воно служить для позначення безлічі взаємодіючих з певною метою об'єктів у загальному контексті системи, що моделюється. Мета самої кооперації у тому, щоб специфікувати особливості реалізації окремих найбільш значущих операцій у системі. Кооперація визначає структуру поведінки системи у термінах взаємодії учасників цієї кооперації.

Діаграма кооперації в нотатції UML представлена рис. 3.8.

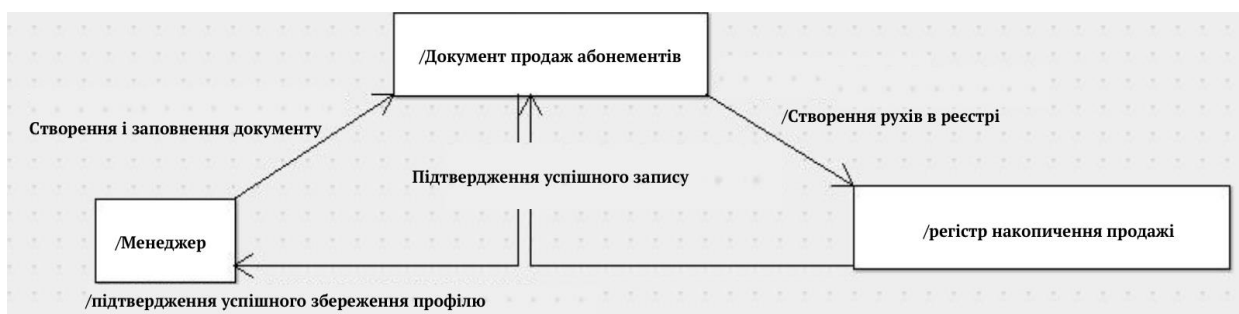


Рис. 3.8. Діаграма кооперації

У цій системі виділено 3 об'єкти: документ «Продаж абонементів», «Менеджер», реєстр накопичення «Продажі». Також було визначено зв'язки між об'єктами.

Діаграма розгортання – це діаграма, де представлені вузли виконання програмних компонентів реального часу, і навіть процесів та об'єктів [17].

Діаграма розгортання застосовується для представлення загальної конфігурації та топології розподіленої програмної системи та містить зображення розміщення компонентів за окремими вузлами системи [17]. З іншого боку, діаграма розгортання показує наявність фізичних сполук – маршрутів передачі між апаратними пристроями, задіяними у реалізації системи. Діаграма розгортання представлена рис. 3.9.

Діаграма розгортання призначена для візуалізації елементів та компонентів програми, що існують лише на етапі її виконання.

Виділяємо основні компоненти автоматизованої інформаційної системи обліку діяльності фітнес-клубу.

Інтерфейс користувача включає сукупність програмних і апаратних засобів, призначених для забезпечення взаємодії між користувачем і програмою, а також зовнішній вигляд програми: розташування елементів, загальний дизайн. Від якості інтерфейсу складається загальне враження роботи програми. Саме тому його розробці необхідно приділяти особливу увагу.

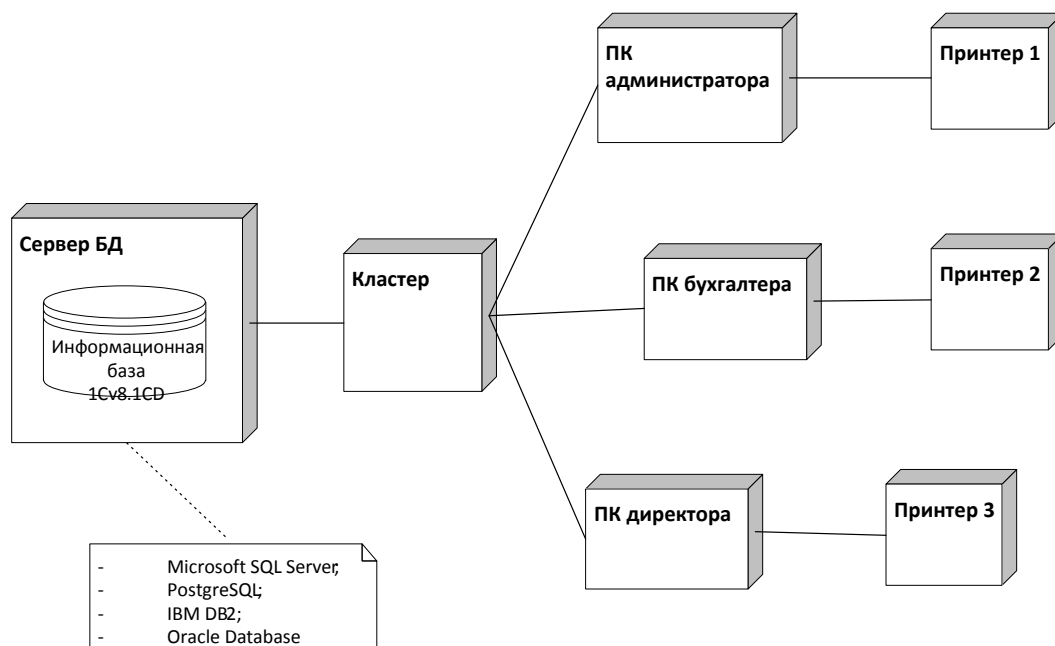


Рис. 3.9. Діаграма розгортання

На даний момент майже кожен бізнес має CRM-систему. Це зручно для бізнесу, надає чудову можливість швидко залучати до роботи нових робітників та переглядати їх ефективність роботи, в пару кліків розсилати акційні пропозиції для існуючих клієнтів та користуватись аналітикою для всіх напрямків роботи бізнесу. Але збереження даних в єдиній системі, тим паче коли вона запозичена, а не створена виключно для однієї компанії - чи є це безпечно?

Розглянемо комерційні пропозиції з безпеки даних в двох найактуальніших CRM-системах, таких як NetHunt CRM та Perfectum, проаналізувавши їх.

Засади безпеки даних NetHunt CRM

«Ви — власник даних CRM

NetHunt ані зберігає інформацію, яку ви додаєте у наш продукт, ані є її власником. Права власності на свої дані є лише у вас.

Доступ до Gmail API через NetHunt CRM обмежується інформацією, необхідною для впровадження програми. Ви можете переглядати та видаляти дозволи для свого облікового запису.

Підтверджена безпека даних. Ми успішно пройшли першу оцінку безпеки від Google у 2019 році, і тепер щороку її проходимо, щоб підтвердити нашу відповідність всім стандартам безпеки.

Дотримання конфіденційності. NetHunt створено відповідно до правил щодо конфіденційності та захисту даних. Це включає GDPR у ЄС та CCPA.

Функції безпеки. Наші функції безпеки допомагають захистити вас від несанкціонованого доступу та контролювати доступ користувачів до даних.

Захист від видалення даних. Папки, записи та навіть поля, видалені вашими користувачами, зберігаються в кошику протягом 30 днів і є доступними для відновлення.

Інфраструктура Google Cloud. Ваші дані NetHunt CRM зберігаються на платформі Google Cloud. NetHunt робить зашифровані резервні копії даних для цілей відновлення.

Зашифровані дані. NetHunt шифрує дані, які передаються між нашими об'єктами, гарантуючи, що до них можуть отримати доступ лише авторизовані ролі.»

Засади безпеки даних Perfectum:

«Perfectum CRM+ERP забезпечує безпеку і конфіденційність даних відповідно до міжнародних стандартів ISO.

Шифрування системи. Усі службові файли Perfectum CRM+ERP зберігаються в повністю зашифрованому вигляді за найсучаснішою технологією, на даний момент (в цілях безпеки не можемо вказати), що надає додатковий захист як від зовнішніх, так і від внутрішніх втручань в роботу системи.

Модуль безпеки. Вбудований модуль безпеки дозволяє обмежити доступ до системи по IP, типам пристроїв, вказати час життя сесії і зашифрувати шлях в систему.

Ролі та дозволи. Гнучка система прав доступу і ролей Perfectum CRM+ERP дозволяє вибудувати чітку ієрархію з урахуванням організаційної структури компанії.

Права доступу. Perfectum CRM+ERP здатна задовольнити будь-які вимоги щодо розмежування доступу до розділів і модулів в системі.

Захист пароля. Функціонал системи дозволяє встановлювати необхідну складність пароля. Паролі хешируються і не зберігаються у відкритому вигляді.

Журнал операцій. У системний журнал записуються всі операції, що відбуваються в системі. Це дозволяє проводити моніторинг всіх подій.

Якщо провести детальну аналітику заявленої інформації від розробників, то можна зробити певні висновки і одразу визначити слабкі місця систем.

Аналіз безпеки даних в NetHunt CRM, заснований на наданій інформації, дозволяє визначити різні аспекти, які спрямовані на забезпечення найвищого рівня захисту для користувачів. Нижче розглянемо ключові елементи безпеки в контексті наданої інформації:

Власність та Контроль Даних: NetHunt CRM поінформував, що вони не зберігають інформацію, яку власник додає у продукт, і не є власником цієї інформації. Це важливо, оскільки це підкреслює принцип власності та контролю користувача над власними даними.

Права Доступу: Інформація про обмеження доступу до Gmail API вказує на уважний підхід до безпеки. Можливість перегляду та видалення дозволів для свого облікового запису дозволяє користувачам контролювати доступ до своїх даних через NetHunt CRM.

Перевірена Безпека Даних: Факт успішної першої оцінки безпеки від Google та регулярні оцінки для підтвердження відповідності стандартам безпеки свідчать про серйозний підхід компанії до захисту даних користувачів.

Дотримання Конфіденційності: Заявлене дотримання правил конфіденційності та захисту даних, таких як GDPR в ЄС та CCPA, свідчить про глобальну орієнтацію NetHunt CRM на важливість забезпечення конфіденційності інформації.

Функції Безпеки: Наявність функцій безпеки, спрямованих на захист від несанкціонованого доступу та контроль доступу користувачів до даних, свідчить про те, що NetHunt CRM враховує сучасні тенденції у сфері кібербезпеки.

Захист від Видалення Даних: Важливість захисту видалених даних через зберігання їх у кошику протягом 30 днів є доповненням до політики відновлення та захисту даних.

Інфраструктура Google Cloud та Зашифровані Дані: Використання інфраструктури Google Cloud для зберігання даних свідчить про високий ступінь надійності та безпеки. Зашифрування передаваних даних гарантує конфіденційність та цілісність інформації. Але в одночас з цим треба зрозуміти, що на серверах Google зберігається дуже багато важливих даних різних компаній та приватних осіб, що дає чітке розуміння регулярних кібератак на сервери Google та дані, розміщені на них. У порівнянні з локальною мережею данних питання безпеки даної CRM сильно програє.

Аналіз Безпеки Даних в Perfectum CRM+ERP:

Відповідність міжнародним стандартам: Заява про забезпечення безпеки та конфіденційності даних відповідно до міжнародних стандартів ISO свідчить про прагнення компанії до високих стандартів безпеки.

Шифрування системи: Використання повністю зашифрованого вигляду для службових файлів, над якими взято верхнє шифрування, є важливим елементом безпеки. Невказання конкретної технології шифрування може бути стратегією захисту, але важливо забезпечити її сучасністю та відповідністю стандартам.

Модуль безпеки: Інтегрований модуль безпеки, який дозволяє обмежити доступ за IP, типами пристроїв, вказати час сесії та зашифрувати шлях в систему, вказує на уважний підхід до контролю доступу.

Ролі та дозволи: Гнучка система прав доступу та ролей сприяє створенню ієрархії з урахуванням організаційної структури компанії. Це важливо для забезпечення обмеженого доступу для різних користувачів.

Права доступу: Можливість задовольнити будь-які вимоги щодо розмежування доступу свідчить про гнучкість системи в управлінні правами доступу.

Захист пароля: Можливість встановлення складних паролів та хешування паролів є ключовими елементами безпеки. Це дозволяє захистити інформацію від несанкціонованого доступу.

Журнал операцій: Запис всіх операцій у системний журнал гарантує можливість моніторингу всіх подій та вчасного виявлення будь-яких відхилень.

Узагальнюючи, Perfectum CRM+ERP виходить із заявою про використання сучасних методів шифрування, гнучкої системи прав доступу та інших заходів безпеки для захисту конфіденційності та цілісності даних. Важливо, що компанія акцентує на відповідність міжнародним стандартам ISO, що свідчить про серйозність її підходу до захисту даних.

Підсумувавши отриману інформацію можна зробити один логічний висновок: В умовах постійного росту кількості цифрових загроз та зростання важливості збереження конфіденційності даних, створення власної CRM-системи із розташуванням на локальному сервері може бути розглянуто як один із найкращих варіантів для забезпечення безпеки.

Локальний Сервер: Розташування CRM-системи на локальному сервері дозволяє повністю контролювати фізичний доступ до серверних пристроїв. Це зменшує ризик витоку даних через мережу та покращує загальний рівень безпеки.

Система Контролю Доступів: Власна CRM дозволяє впровадження високорівневої системи контролю доступу. Задавання різних рівнів доступу для різних користувачів зменшує ймовірність несанкціонованого доступу до конфіденційної інформації.

Журнал Подій: Важливим елементом безпеки є ведення журналу подій. Журналювання дозволяє відстежувати всі події та операції, що відбуваються

у системі. Це може служити не тільки для виявлення можливих загроз, але й для відновлення даних в разі необхідності.

Захист Від Видалення Даних: Власна CRM може надати можливість встановлення заходів захисту видалених даних, зберігаючи їх на певний термін та надаючи можливість відновлення.

Кастомізація та Оновлення: Власна система дозволяє вам гнучко налаштувати систему з урахуванням унікальних потреб вашого бізнесу. Регулярні оновлення дозволяють підтримувати безпеку на високому рівні.

Хоча створення власної CRM-системи вимагає значних витрат часу та ресурсів, це інвестиція в довгостроковий захист ваших даних. Наявність контролю, журналування подій та інших елементів безпеки дозволяє більш ефективно реагувати на виклики кібербезпеки та зберігати високий рівень конфіденційності корпоративної інформації.

3.3. Планування розробки та оцінка бюджету

Управління проектами – це сукупність процесів із планування, координації та контролю робіт для реалізації цілей проектів з урахуванням обмежень на ресурси, бюджет та вимог якості. У ширшому розумінні під керівництвом проектами розуміється застосування знань, практичного досвіду, інструментальних засобів і методів задоволення потреб зацікавлених осіб проекту [20].

У ході планування проекту було виділено 20 завдань. Список завдань проекту, їх види, тривалість та попередники наведені в табл. 3.5.

У MS Project 2013 створимо проект із початковою датою 13.10.2023. Занесемо вище перелічені завдання у створений проект.

Виділимо необхідні ресурси для виконання кожного із завдань даного проекту. Ресурси та витрати представлені в табл. 3.6.

Завдання проекту

№	Назва	Вид Завдання	Попередники	Тривалість
1	Початок проекту	Віха	-	-
2	Аналіз предметної галузі	Фаза	-	-
3	Перегляд ринку інформаційних систем	Завдання	-	2
4	Збір інформації про існуючі інформаційні системи	Завдання	3	3
5	Аналіз вимог	Фаза	-	-
6	Аналіз функціональних вимог	Завдання	4	6
7	Аналіз використовуваних технологій	Завдання	6	4
8	Проектування	Фаза	-	-
9	Проектування базової архітектури	Завдання	6;7	2
10	Проектування логічної моделі бази даних	Завдання	6	2
11	Проектування інтерфейсу користувача	Завдання	6;7	7
12	Проектування програмних засобів	Завдання	7;11	8
13	Реалізація	Фаза	-	-
14	Реалізація фізичної моделі бази даних	Завдання	10	2
15	Реалізація клієнтської частини програми	Завдання	9; 11; 12; 14	14
16	Тестування	Фаза	-	-
17	Функціональне тестування	Завдання	15	7
18	Завершення проекту	Фаза	-	-
19	Оформлення документації	Завдання	18	7
20	Кінець проекту	Віха	-	-

Ресурси проекту

Назва ресурсу	Тип	Витрати		
		Таблиця норм	Станд . Ставка	Ставка понадур .
Програміст 1	Т	А	24000 грн. / міс	350 грн. / год
Програміст 2	Т	А	22000 грн. / міс	300 грн. / год
Аналітик	Т	А	17000 грн. / міс	250 грн. / год
Тестувальник	Т	А	16000 грн. / міс	250 грн. / год
Комп'ютер	М	А	15000 грн.	
Принтер	М	А	4000 грн.	
Папір	М	А	250 грн.	
Інтернет	З			

Розподіл ресурсів із завдань представлено у таблиці 3.7.

Розподіл ресурсів із завдань

Назва	Ресурс	Одиниці (Витрати)	Таблиця норм витрат
Початок проекту	Комп'ютер	4	А
Аналіз ринку інформаційних систем	Аналітик, Інтернет	100	А 250
Збір інформації про існуючі інформаційні системи	Аналітик, Інтернет	100	А 350
Аналіз функціональних вимог	Аналітик, Програміст 1	100 100	А А
Аналіз використовуваних технологій	Програміст 1, Аналітик	100 30	А А
Проектування базової архітектури	Програміст 1	100	А
Проектування логічної моделі бази даних	Програміст 2	50	А
Проектування інтерфейсу користувача	Програміст 2	50	А
Проектування програмних засобів	Програміст 1	100	А

Реалізація фізичної моделі бази даних	Програміст 2	50	A
Реалізація частини клієнтської програми	Програміст 1, Програміст 2	100 100	A A
Функціональне тестування	Тестувальник, Програміст 1	70 30	A A
Оформлення документації	Програміст 1, Програміст 2 Тестувальник, Принтер, Папір	50 100 30 1 1	A A A A A

Занесемо вище перелічені ресурси у проект. Результати представлені рис. 3.10.

	Название ресурса	Тип	Единицы измерения материалов	Краткое название	Макс. единиц	Стандартная ставка	Ставка сверхурочных	Начисление	Базовый календарь
1	Программист1	Трудовой		Прог1	100%	24 000,00р./мес	350,00р./ч	Пропорциональное	Стандартный
2	Программист2	Трудовой		Прог2	100%	22 000,00р./мес	300,00р./ч	Пропорциональное	Стандартный
3	Аналитик	Трудовой		Аналит	100%	17 000,00р./мес	250,00р./ч	Пропорциональное	Стандартный
4	Тестирующк	Трудовой		Тестир	100%	16 000,00р./мес	250,00р./ч	Пропорциональное	Стандартный
5	Компьютер	Материальный	Штука	Комп		15 000,00р.		Пропорциональное	
6	Принтер	Материальный	Штука	Принт		4 000,00р.		Пропорциональное	
7	Бумага	Материальный	Пачка	Бум		250,00р.		Пропорциональное	
8	Интернет	Затраты		Инет				Пропорциональное	

Рис. 3.10. Ресурси проекту

Далі закріпимо за завданнями відповідні ресурси. Після, MS Project запропонує наочну ілюстрацію календарного плану у вигляді діаграми Ганта. Отримана діаграма Ганта наведено рис. 3.11. Мережевий графік даного проекту представлено рис. 3.12.

З табл. 3.8 видно, що тривалість виконання проекту становила 54 дні, яке бюджет становив 189 010 грн.



Рис. 3.11. Діаграма Ганта

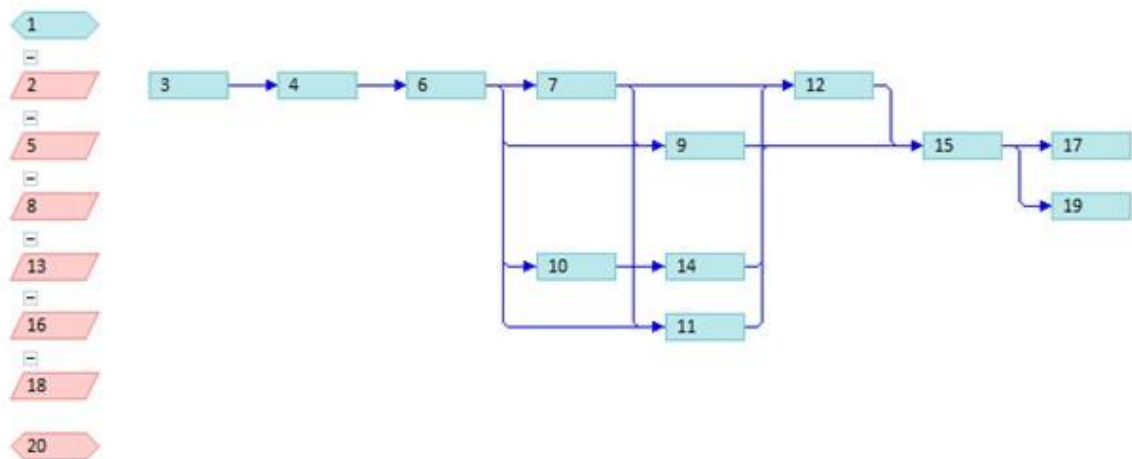


Рис. 3.12.– мережевий графік

У процесі виконання випускної кваліфікаційної роботи було виконано аналіз предметної області, у межах якого було проведено огляд існуючих систем автоматизації діяльності фітнес-клубу, завдяки чому було виявлено структуру та основні функції конфігурації, що розробляється. Крім цього, виконано аналіз функцій та бізнес-процесів підприємства. Також було проведено аналіз системи з допомогою різних діаграм UML : діаграма варіантів використання, діаграма станів, діаграма діяльності, діаграма послідовності, діаграма кооперації. Наведено опис вибраних технологій та засобів розробки, а також обґрунтовано їх вибір.

Виконання проєкту

Початок проєкту	0,00 ⁰
Аналіз предметної області	10000,00⁵
Перегляд ринку інформаційних систем	4000,00 ²
Збір інформації про існуючі інформаційні системи	6000,00 ³
Аналіз вимог	38400,00
Аналіз функціональних вимог	26400,00
Аналіз технологій, що використовуються	12000,00
Проектування	33000,00
Проектування базової архітектури	4800,00
Проектування логічної моделі БД	2000,00
Проектування інтерфейсу користувача	7000,00
Проектування програмних засобів	19200,00
Реалізація	63600,00
Реалізація фізичної моделі БД	2000,00
Реалізація клієнтської частини додатку	61600,00
Тестування	13664,00
Функціональне тестування	13664,00
Завершення проєкту	30346,00
Оформлення документації	30346,00
Кінець проєкту	0,00

В результаті було розроблено конфігурацію для автоматизації діяльності фітнес-клубу, яка дозволяє вести звітність і контролювати бізнес-процеси, а також вести ефективні взаємини з клієнтами.

Тестування програмних засобів полягало у визначенні методу тестування та виконання тестів. Результати функціонального тестування показали коректність конфігурації.

Розроблений прототип надалі може бути вдосконалено використанням у конфігурацію додаткових функцій.

Висновки до третього розділу

У ході створення автоматизованої інформаційної системи для фітнес-клубу, використовуючи CASE-засіб ERwin Data Modeler та інші інструменти, було проведено проектування бази даних та розроблено бізнес-процеси, пов'язані з діяльністю клубу. Вхідні дані системи включають інформацію про клієнтів, відвідувачів та тарифні плани.

Для оптимізації діяльності фітнес-клубу було визначено ключові бізнес-процеси, такі як "Облік клієнтів", "Продаж абонементів", "Відвідування фітнес-клубу", "Облік заробітної плати працівникам", "Складання графіка занять" та "Формування звітів". Ці процеси були відображені на декомпозиційній діаграмі та діалізуючій діаграмі в нотаціях IDEF0.

Окрім того, було проведено проектування програмного забезпечення для ефективного виконання бізнес-процесів. Детальний аналіз безпеки даних в NetHunt CRM та Perfectum був виконаний, щоб забезпечити надійність та конфіденційність інформації, зокрема документів про надані послуги, звітів, нарахувань заробітної плати та інших даних.

У результаті виконаних кроків можна зробити висновок, що створена інформаційна система для фітнес-клубу є добре структурованою та функціональною, спрямованою на покращення управління клубом, оптимізацію бізнес-процесів та забезпечення безпеки даних.

ВИСНОВКИ

Безпека баз даних в сучасному бізнес-середовищі відіграє ключову роль у забезпеченні надійності, цілісності та конфіденційності важливої інформації. З урахуванням зростаючого обсягу даних та загроз кібербезпеки, дбайливе управління безпекою баз даних стає вирішальною для успіху підприємства. У роботі доведено, чому оптимізація баз даних в бізнесі є невід'ємною частиною стратегії управління ризиками.

Однією з головних причин для врахування оптимізації баз даних є потреба в захисті конфіденційності даних. У бізнесі важлива інформація, така як особисті дані клієнтів, фінансові дані чи комерційні відомості, вимагає особливого рівня захисту. Зловмисники можуть використовувати недоліки в безпеці для несанкціонованого доступу, що може призвести до втечі конфіденційної інформації та завдати серйозного збитку репутації компанії.

Цілісність даних є важливою частиною довіри до інформації. Якщо дані можуть бути змінені без авторизації чи сліду, це може призвести до неправильних вирішень та втрати довіри. Забезпечення цілісності баз даних включає в себе встановлення механізмів контролю доступу та моніторингу, які унеможливають несанкціоновані зміни.

Втрати даних можуть стати серйозним викликом для будь-якої компанії. Відновлення втрачених даних може виявитися складним і витратним процесом. Регулярні резервні копії та механізми відновлення є ключовими для запобігання втратам даних. Додатково, розробка та виконання стратегій реагування на інциденти дозволить ефективно вирішувати проблеми, коли вони виникають.

Багато галузей бізнесу підпадають під специфічні регуляторні вимоги, щодо яких вимагається дотримання. Забезпечення безпеки баз даних важливо для відповідності цим нормам. Наприклад, у сфері охорони здоров'я чи

фінансових послуг існують високі стандарти безпеки, які компанії повинні дотримуватись.

В бізнесі, де інформація є ключовим активом, безпека баз даних не може бути недооціненою. Захист конфіденційності, цілісності та доступності даних є критичним для успішної діяльності підприємства та збереження довіри клієнтів. Інвестування в безпеку баз даних – це не просто заходи безпеки; це стратегічна ініціатива, яка сприяє стабільності та довгостроковому успіху компанії в цифровому світі.

Використавши усі розроблені в даній дипломній роботі системи оптимізації даних Ви можете максимально захистити всю інформаційну складову Вашого бізнесу, але слід не забувати, що оптимізація даних розробляється станом на сьогодні.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Гайдаржи В. Бази даних в інформаційних системах
2. Бази даних та етапи їх створення. URL: <https://ppt-online.org/496817>
3. Дослідження міжнародного кадрового порталу Headhunter Україна
4. Дрейк Джошуа Д. Practical PostgreSQL. 2002
5. Бейлі Л. Head First SQL - Your brain on SQL. 2007
6. Дейт К. An Introduction to database systems, 1975
7. Петров А. Database internals: a deep dive into how distributed database works. 2019
8. Тарасов С. СКБД для програміста - Бази даних з середини»
9. Відкриття системи. СУБД. *Кібербезпека*, 2022
10. Кемпбелл Л., Мейджорс Ч. Бази даних - інжинірінг надійності, 2020
11. Альмухаметов В. Розробка і проектування СУБД. 2019
12. Сьоре Е. Проектування та реалізація систем управління базами даних., 2021
13. Новіков Б. О., Горшкова Є. О., Графєєва Н. Г. Основи технологій баз даних. 2004
14. Рядовий приклад CRM-системи URL: <https://www.terrasoft.ua/page/crm-definition>
15. Грінберг П. CRM зі швидкістю світла. Залучення і утримання клієнтів в реальному часі через інтернет
16. Кінзябулатов Р. CRM детально і по справі
17. Миронюк К. Створи і запусти прибутковий SaaS бізнес
18. Рязанцев О. Як запровадити CRM-систему за 50 днів
20. Марк У. Джонстон, Грег У. Маршалл. Управління відділом продажів
21. Структура даних. Класифікація структур даних. URL: <http://um.co.ua/6/6-11/6-113691.html>

22. 11 типів сучасних баз даних: короткий опис, схеми і приклади БД. URL: <https://senior.ua/articles/11-tipiv-suchasnih-baz-danikh-korotkiy-opis-shemi--prikлади-bd>
23. Введення в сучасні бази даних. URL: <http://ir.nmu.org.ua/bitstream/handle/123456789/154887/MA+Demidenko+INTRODUCTION+TO+MODERN+DATABASES.pdf?sequence=1&isAllowed=y>
24. Сучасні БД. URL: <https://lektsia.com/14x355.html>
25. Системна архітектура «клієнт-сервер». URL: <https://studfile.net/preview/9250282/>
26. Класифікація структур даних. URL: https://life-prog.ru/ukr/view_algoritmleng.php?id=55
27. Комп'ютерні технології використання систем управління базами даних на окремих АРМ і в обчислювальних мережах. URL: <http://um.co.ua/6/6-17/6-17910.html>
28. Системи обробки економічної інформації. URL: https://ua-referat.com/Режим_роботи_з_базами_даних
29. Бази даних. Етапи створення бази даних. URL: <https://present5.com/bazi-danix-etapi-stvorennnya-bazi-danix-ponyattyа-pro/>
30. Технологія використання СУБД. URL: https://studopedia.com.ua/1_16807_tehnologiya-vikoristannya-subd.html
31. Теоретичні основи функціонування баз даних. URL: <http://samzan.net/5361>
32. СКБД. URL: <https://miyklas.com.ua/p/informatica/9-klas/bazi-danikh-sistemi-keruvannia-bazami-danikh-361840/sistemi-keruvannia-bazami-danikh-352450/re-b0f14256-7ac2-4823-9f53-656061b85eaa>
33. База Даних. URL: https://uk.wikipedia.org/wiki/База_даних
34. Схема даних та технологія керування його об'єктами. URL: <https://studfile.net/preview/5015570/>
35. Структури баз даних. URL: <https://infopedia.su/12x12a3.html>

36. Фінансово-кредитна система України в умовах інтеграційних і глобалізаційних процесів. URL: https://cibs.ubs.edu.ua/wp-content/uploads/2019/06/materialy_11042019.pdf
37. Проектування баз даних. URL: <https://shag.com.ua/1-osnovni-ponyattya-baz-danih-proektuvannya-baz-danih.html?page=2>
38. Порядок дій при проектуванні баз даних. URL: <https://uroki5.ru/informatika/question41156821>
39. Елементи бази даних. URL: <https://marinaintel111.blogspot.com/2018/10/blog-post.html>
40. Побудова СКБД. URL: https://ela.kpi.ua/bitstream/123456789/28779/1/Bondarenko_bakalavr.pdf
41. Безуглий Д. Інформаційна безпека України: огляд останніх тенденцій. Фізико-математична освіта. 2018. Вип. 2(16). С. 13–17.
42. Державна служба статистики України. Київ, 2022. URL: <http://www.ukrstat.gov.ua> (дата звернення: 20.03.2022).
43. Український інститут майбутнього. Київ, 2022. URL: <http://www.uifuture.org> (дата звернення: 19.04.2022).
44. Міністерство та Комітет цифрової трансформації України. Київ, 2020. URL: <https://thedigital.gov.ua/news/ukhvaleno-zakon-pro-zakhist-informatsii-v-telekomunikatsiy-nikh-sistemakh> (дата звернення 13.10.2021).
45. Ajay Ohri. Importance of Data Security In 2021: Jigso Academy. 20 February, 2021. URL: <https://www.jigsawacademy.com/blogs/cyber-security/importance-of-data-security/>
46. Harrington D. Data Security: Definition, Explanation and Guide: Varonis / Inside Out Security Blog. July 6, 2021. URL: <https://www.varonis.com/blog/data-security/>
47. Data Security. Imperva, 2021. URL: <https://www.imperva.com/learn/data-security/data-security/>