

Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи

магістра

(назва освітньо-кваліфікаційного рівня)

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальність 125 Кібербезпека
(код і назва спеціальності)

освітній рівень магістр
(назва освітнього рівня)

кваліфікація _____
(код і назва кваліфікації)

на тему: Інформаційно-аналітична система вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів

Виконавець: студент 2 курсу, групи КБм-21

Андрущенко Ярослав Вячеславович

(підпис)

(прізвище ім'я по-батькові)

	Прізвище, ініціали	Оцінка	Підпис
Науковий керівник	Бучік С.С.		
Рецензент	Кудін В.І.		
Нормоконтроль	Фесенко А.О.		

Київ
2021

Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:

завідувач кафедри
кібербезпеки та захисту інформації

_____ Лукова-Чуйко Н.В.

«_____» _____ 20__ року

ЗАВДАННЯ

на виконання дипломної роботи

спеціальності _____

125 Кібербезпека

(код і назва спеціальності)

студенту _____

КБм-21

(група)

Андрущенко Ярославу Вячеславовичу

(прізвище ім'я по-батькові)

Тема дипломної роботи _____

*Інформаційно-аналітична система вибору та
аналізу комплексу засобів захисту державних інформаційних ресурсів*

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № _____ від _____

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБИТ

Об'єкт досліджень *Процес вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.*

Предмет досліджень *Елементи інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.*

Мета _____

*Розробити елементи інформаційно-аналітичної системи
вибору та аналізу комплексу засобів захисту
державних інформаційних ресурсів*

Вихідні дані для проведення роботи *Офіційні переліки дозволених засобів захисту державних інформаційних ресурсів.*

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна Зменшується час на прийняття рішень щодо застосування певних засобів захисту інформації при створенні КСЗІ та збільшується ефективність таких рішень.

Практична цінність Зменшується час на прийняття рішень щодо застосування певних засобів захисту інформації при створенні КСЗІ та збільшується ефективність таких рішень.

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практичне доведення ефективності створення інформаційно-аналітичної системи, як інструменту для експертів з побудови КСЗІ

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Уточнення постановки задачі	20.09.2020-19.10.2021
Аналіз літературних джерел	22.10.2020-14.05.2021
Обґрунтування вибору рішення	15.10.2021-25.20.2021
Збір інформації	26.09.2020-14.05.2021
Аналіз даних та їх класифікація	20.09.2020-19.05.2021
Визначення особливостей побудови інформаційно-аналітичної системи	20.09.2020-19.05.2021
Вдосконалення обробки та зберігання інформації ІАС	20.09.2020-19.05.2021
Створення програмної реалізації побудови бази даних та програмного забезпечення	20.09.2020-19.05.2021
Оформлення друку та пояснювальної записки	20.04.2021-25.05.2021

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект Експерт, що обирає засоби для захисту державних інформаційних ресурсів, працюватиме ефективніше.

Зменшується необхідна кваліфікація експерта для

Соціальний ефект прийняття зваженого рішення при виборі засобів захисту державних інформаційних ресурсів.

7. ДОДАТКОВІ ВИМОГИ

Завдання видав	_____	_____
	(підпис)	(прізвище, ініціали)
Завдання прийняв до виконання	_____	_____
	(підпис)	(прізвище, ініціали)

Дата видачі завдання: _____
Термін подання дипломної роботи до ЕК _____

УДК [004.4+004.65](004.91)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Розробка елементів інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів»: 101 сторінок, 17 рисунків, 4 додатки та 9 таблиць, 19 літературних джерела.

Об'єкт дослідження – процес вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Мета роботи – розробити елементи інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Предмет дослідження – елементи інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Методи дослідження – методи експертного оцінювання, метод аналізу і синтезу.

У роботі проведено аналіз офіційних переліків дозволених засобів захисту державних інформаційних ресурсів, на основі яких запропонована інформаційно-аналітична система, а також практично втілені її окремі елементи, а саме: розроблено базу даних дозволених засобів та прототип програмного забезпечення для проведення експертизи щодо засобів захисту та виводу кінцевого переліку вибраних засобів.

Практична значимість: полегшено процес відбору засобів захисту інформації з переліків дозволених засобів захисту державних інформаційних ресурсів при створенні комплексної системи захисту інформації, шляхом зменшення часових витрат на їх відбір та зменшення вимог до кваліфікації експерта, що приймає кінцеве рішення.

Актуальність теми: через недоліки в укладанні офіційних переліків дозволених засобів захисту державних інформаційних ресурсів та незручність їхнього використання необхідно провести їх реструктуризацію та створити інформаційно-аналітичну систему для підвищення зручності їх використання.

Напрямок подальших досліджень: втілення та вдосконалення структури інформаційно-аналітичної системи, запропонованій у роботі.

Ключові слова: інформаційно-аналітична система, дозволені засоби технічного захисту інформації, державні інформаційні ресурси, комплексна система захисту інформації, захист інформації, технічний захист інформації, база даних.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ.....	8
ВСТУП.....	9
РОЗДІЛ 1 АНАЛІЗ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ В КІБЕРБЕЗПЕЦІ ТА ІНФОРМАЦІЙНІЙ БЕЗПЕЦІ. АКТУАЛЬНІСТЬ РОБОТИ.....	13
1.1. Дослідження предметної області.....	13
1.2. Аналіз інформаційно-аналітичних систем та систем підтримки прийняття рішень, використовуваних у сфері кібербезпеки.....	29
1.3. Аналіз поточного стану вибору засобів захисту інформації.....	37
1.4. Актуальність створення бази даних та аналітичної системи вибору.....	37
Висновок до першого розділу.....	40
РОЗДІЛ 2 РОЗРОБКА СТРУКТУРИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ТА ОКРЕСЛЕННЯ ЇЇ МАЙБУТНЬОГО ФУНКЦІОНАЛУ.....	41
2.1 Огляд функціоналу інформаційно-аналітичної системи.....	41
2.2 Структура інформаційно-аналітичної системи.....	48
2.3 Визначення елементів для розробки.....	52
Висновки до другого розділу.....	53
РОЗДІЛ 3 РОЗРОБКА КОМПОНЕНТІВ ТА БАЗИ ДАНИХ ЯК ЕЛЕМЕНТУ ВХІДНИХ ДАНИХ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ.....	54
3.1 Розробка бази даних. Інфологічна модель.....	54
3.2 Розробка бази даних. Даталогічна модель.....	60
3.3 Розробка бази даних. Втілення моделі на сервер реляційної бази даних.....	68
Висновки до третього розділу.....	77

РОЗДІЛ 4 ДОСЛІДЖЕННЯ РОБОТИ ПРОТОТИПУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВВОДУ ДАНИХ ТА ПРЕДСТАВЛЕННЯ РЕЗУЛЬТАТІВ.....	78
4.1 Обґрунтування вибору мови і системи програмування.....	78
4.2 Обґрунтування вибору середовища програму.....	80
4.3 Опис програми та її алгоритми.....	80
4.3.1 Панель адміністратора.....	80
Висновки до четвертого розділу.....	84
ВИСНОВКИ.....	85
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	87
ДОДАТКИ.....	90
Додаток А. Приклад коду створення бази даних ІАС.....	90
Додаток Б. Приклад коду створення бази даних ІАС.....	91
Додаток В. Приклад коду для побудови рекомендацій для експерту.....	94
Додаток Г. Програма керуванням та фільтрації таблиць за прикладом таблиці “Засоби”.....	97

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

БД – база даних

ДССЗІ – Державна служба спеціального зв'язку та захисту інформації України

ЗЗІ – засіб захисту інформації

ІАС – інформаційно-аналітична система

КСЗІ – комплексна система захисту інформації

НД ТЗІ – Нормативний документ системи технічного захисту інформації

ПЗ – програмне забезпечення

СППР – система підтримки прийняття рішень

ТЗІ – технічний захист інформації

ТПКО – Тимчасове положення про категорювання об'єктів

ВСТУП

Комплексна система захисту інформації – взаємопов’язана сукупність організаційних та інженерно-технічних заходів, методів і заходів безпеки інформації [1]. Побудова комплексної системи захисту інформації є необхідною умовою для забезпечення усіх властивостей безпеки інформації (конфіденційності, цілісності та доступності); захист інформації у державних установах поза КСЗІ неможливий.

Згідно з інформації про Положенням «Про технічний захист інформації в Україні, «під час виготовлення і впровадження заходів та методів з технічного захисту інформації мають бути використані засоби, дозволені Адміністрацією Державної служби спеціального зв’язку та захисту інформації України для застосування та включені до відповідних переліків» [2]. Тобто КСЗІ дозволено будувати лише з тих компонентів, що мають відповідний сертифікат чи експертний висновок від ДССЗІ. Переліки таких засобів публікуються на офіційному сайті ДССЗІ – проте формат, у якому вони подані, незручний для опрацювання через великий обсяг і одночасно регулярні оновлення, погану класифікацію засобів захисту інформації та відсутність єдиного документу з переліком дозволених засобів технічного захисту інформації.

Для того, аби прийняти адекватне рішення щодо створення КСЗІ, потрібно не тільки з точки зору кваліфікованого експерта оцінити наявні ресурси і потреби в захисті, але й знати зміст нормативних документів у сфері створення КСЗІ.

Документи (Нормативні документи системи технічного захисту інформації), де зазначені описи рівнів гарантій та функціоналу ЗЗІ (НД ТЗІ 2.5-004-99 «Критерії оцінки захищеності інформації в комп’ютерних системах від несанкціонованого доступу» та НД ТЗІ 2.5-005-99

«Класифікація автоматизованих систем і стандартні функціональні профілі захищеності оброблюваної інформації від несанкціонованого доступу», відповідно), вичерпні, проте містять більше інформації, ніж зазвичай потрібно в момент складання списку компонентів КСЗІ.

Цілком очікувано, що без років досвіду у сфері побудови КСЗІ та докладних знань НД ТЗІ прийняття рішення про складові компоненти буде значною мірою затримане та/або переускладнене.

Судячи з вище перелічених факторів, маємо наступні наявні проблеми:

- незручно побудована таблиця сертифікованих ЗЗІ;
- обраний незручний формат таблиці сертифікованих ЗЗІ;
- завищені вимоги до спеціалістів, залучених до створення КСЗІ на підприємстві.

Ця робота покликана надати початок розробці та втіленню ІАС для комфортного та зваженого прийняття рішення про засоби ТЗІ, використовуваних при створенні КСЗІ, тим самим збільшуючи продуктивність експертів, що приймають остаточне рішення про створення КСЗІ та знижуючи поріг кваліфікації, необхідний для прийняття такого рішення, без зниження якості цього рішення.

Метою роботи є розробка елементів інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Для досягнення зазначеної мети дипломної роботи розроблені та поставлені окремі завдання:

1. Визначити конкретний перелік офіційних документів ДССЗЗІ, де вказані дозволені засоби захисту інформації.
2. Обрати з переліку документи для дослідження.
3. Провести аналіз структури наявних документів, її переваг та недоліків.
4. Визначити, якою інформацією можна доповнити наявну.

5. Розробити структуру інформаційно-аналітичної системи.
6. Продумати функціонал повної інформаційно-аналітичної системи вибору.
7. Розробити інфологічну та даталогічну моделі бази даних, враховуючи інформацію, яку вона міститиме, та зручність її редагування.
8. Втілити даталогічну модель на сервер реляційних баз даних.
9. Розробити інтерфейс інформаційної системи для навігації базою даних як прототип інтерфейсу інформаційно-аналітичної системи.
10. Розробити та протестувати програмний продукт.

Об'єктом дослідження є процес вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Предметом дослідження є елементи інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів.

Методами дослідження є методи експертного оцінювання, метод аналізу та синтезу.

Практичним значенням роботи є:

- отримання зручного представлення офіційних переліків дозволених засобів захисту інформації від ДССЗЗІ;
- зменшення часу та необхідної кваліфікації для оптимального прийняття рішення щодо необхідних засобів захисту для створення КСЗІ за рахунок створення відповідної бази даних;
- удосконалення офіційних переліків дозволених засобів захисту інформації від ДССЗЗІ з точки зору зручності їхнього представлення та користування ними.
- вперше пропонування ДССЗЗІ створення бази даних із розробленою у цій роботі структурою як доповнення до досліджуваних переліків.

Результати роботи доповідалися та обговорювалися на IV міжнародній науково-практичній конференції “проблеми кібербезпеки інформаційно-телекомунікаційних систем”; положення дипломної роботи викладено у науковій праці, викладені в матеріалах конференцій.

РОЗДІЛ 1

АНАЛІЗ ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМ В КІБЕРБЕЗПЕЦІ ТА ІНФОРМАЦІЙНІЙ БЕЗПЕЦІ. АКТУАЛЬНІСТЬ РОБОТИ

1.1 Дослідження предметної області

Дослідження предметної області варто розпочати із визначеннями певних термінів, що будуть використовуватися в даній роботі, адже вони позначають предмети та об'єкти, за яких викристалізувалась ця робота.

Інформаційно-аналітична система (ІАС) — це комп'ютерна система, яка дозволяє збирати, створювати, обробляти і аналізувати інформацію. Цілі МАА полягають в ефективному зберіганні, обробці та аналізі даних. Технологічна платформа ААІ дозволяє компаніям інтегрувати і координувати бізнес-процеси. Типовий ААІ встановлюється в одному або декількох центрах даних, де виконується відповідне корпоративне програмне забезпечення. ІАС може включати додаткові додатки, які працюють в організаційній структурі компанії / підприємств, наприклад, в адміністративних цілях. ААІ надає унікальну інформаційну область для компанії (компаній) та дозволяє використовувати її на всіх функціональних рівнях ієрархії і управління. [1].

База даних дозволених засобів ТЗІ та прототип програмного забезпечення стануть складовими ІАС, що будуть втілені у цій роботі.

База даних — це структурована структура для зберігання, модифікації і обробки взаємозалежної інформації; часто це конкретний набір даних, пов'язаних з великими томами бази даних, загальною властивістю або особливістю, і відсортовано в алфавітному порядку.

Об'єднання великих обсягів даних в єдину базу даних дозволяє змінювати різні інформаційні групи - особисті дані клієнта, історію замовлень, каталог товарів і відпочинок [4].

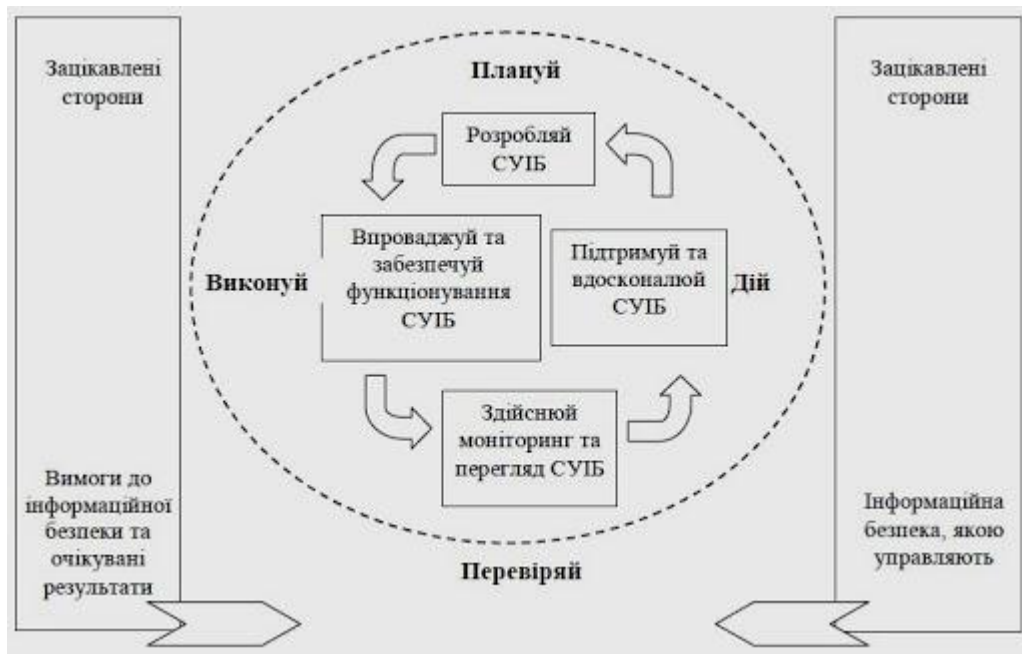


Рисунок 1.1 - Огляд Інформаційно аналітичної системи

Програмне забезпечення – загальна концепція відноситься до набору закодованих інструкцій (програм) для керування процесором комп'ютера (ЦПУ). Процесор комп'ютера зчитує і виконує ці закодовані інструкції. Виконання комп'ютерного програмного забезпечення включає в себе обробку інформації і перевірку компонентів комп'ютерного обладнання. Наприклад, для персональних комп'ютерів типово переглядати і отримувати інформацію з клавіатури. [5].

С самого початку буде побудована інфологічна модель. Базуючись на інфологічній моделі буде створена датологічна модель, яка в свою чергу буде побудована на сервері реляційної бази даних.

Інфологічна модель – ця нова модель відображає реальний світ деяких понять, зрозумілих людині, абсолютно незалежно від зберігання

параметричних даних. Існує багато підходів до створення цих моделей: Сюжет моделі, семантичні мережі, модель "об'єктного з'єднання" і т.д.

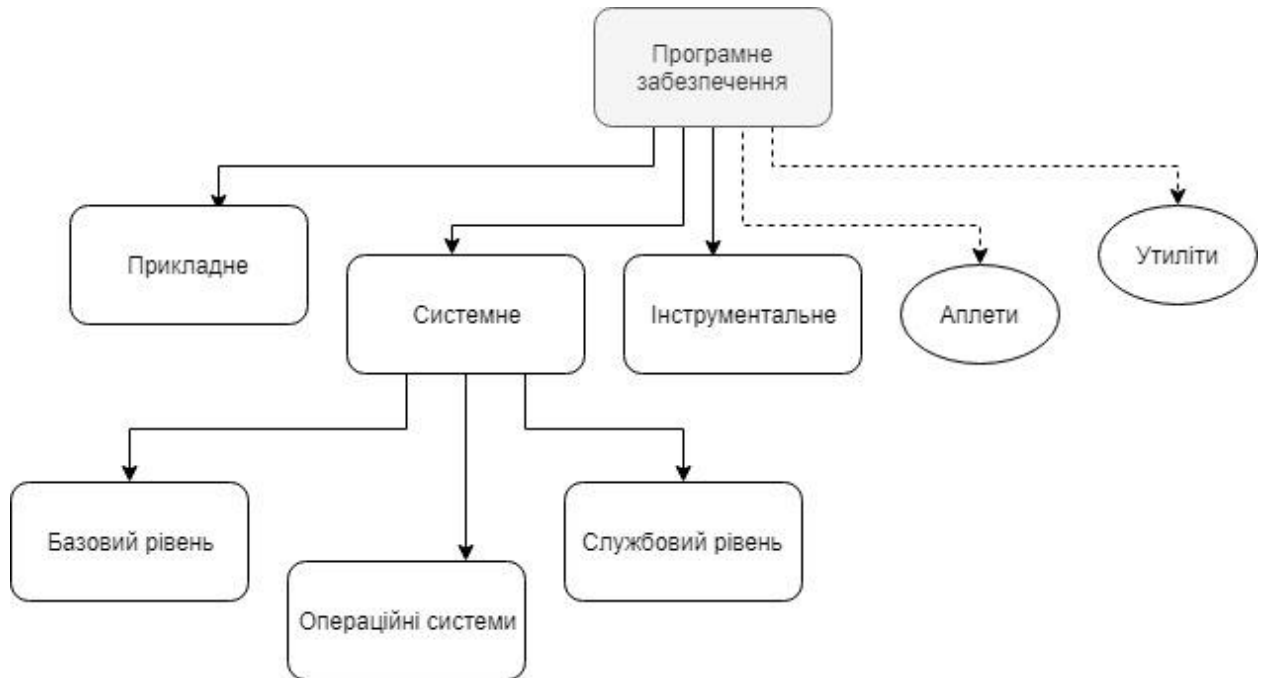


Рисунок 1.2 - Види програмного забезпечення варіант №1

Найбільш сучасні підходи до проектування баз даних (особливо реляційних) засновані на різноманітності моделей ER. Модель була запропонована Ченем в 1976 році. Доменне моделювання ґрунтується на використанні графічних діаграм з невеликим числом різнорідних компонентів. Через прозорості подання концептуальних діаграм баз даних, моделі ER стали загальними в системах ASE, які підтримують автоматичне проектування реляційних баз даних.

Ці елементи являють собою прямокутники, помічені сутностями, ромбами або шестикутниками, асоціаціями (зв'язками), атрибутами, поміченими овалами, і зв'язками між ними. Ступінь зв'язку може бути вказана на (може використовуватися замість 1 або "багато") і вказана необхідним поясненням [6].

Модель системи реєстрації даних є логічно налаштованою базою даних, яка фокусується на конкретній СУБД. Перед виконанням проекту системи

реєстрації даних необхідно вибрати СУБД. Кожна конкретна СУБД має обмеження в структурі логічної моделі даних, так що всі фактори, які можуть вплинути на логічну модель БД, повинні бути визначені в першу чергу, якщо вони не є специфічними для СУБД.



Рисунок 1.3 - Види програмного забезпечення варіант №2

Фактори що впливають проектування датологічної моделі з боку СУБД, є такі:

1. Модель системи реєстрації даних є логічно налаштованою базою даних, яка фокусується на конкретній СУБД. Перед виконанням проекту системи реєстрації даних необхідно вибрати СУБД. Кожна конкретна СУБД має обмеження в структурі логічної моделі даних, так що всі фактори, які можуть вплинути на логічну модель БД, повинні бути визначені в першу чергу, якщо вони не є специфічними для СУБД.

2. Характеристики фізичної організації даних в вибраному середовищі СУБД. Наприклад, в базі даних Paradox або в системі dBASE база даних організована як набір взаємозв'язаних форматів файлів DT і DBF, а всі інші об'єкти, такі як форми і звіти, зберігаються в окремих файлах Lah. В СУБД Міаросóфт Access всі дані і інструменти зберігаються в одному файлі бази даних. Тому при розробці бази даних необхідно знати не тільки правила логічного побудови, але і фізичну організацію бази даних.

3. Кількісні обмеження, що застосовуються СУБД (наприклад, число ієрархічних рівнів, число можливих полів, записів, файлів і т.д.) в ієрархічних моделях [7]



Рис 1.4 Огляд СУБД

Реляційна база даних (рис 1.5) являє собою базу даних, засновану на реляційній моделі, наприклад базу даних БД з методом представлення даних в табличній формі і зовнішню базу даних з однорідними таблицями. Кожен об'єкт записується в рядок в таблиці. Рядок називається реєстрацією. Запис складається з різних типів полів.

Створення реляційної бази даних і управління нею за допомогою спеціальних засобів - реляційних систем управління базами даних.[8].

Реляційна модель даних є логічною моделлю даних. Вперше запропоновано в 1970 році британським вченим Едгаром Франком ХДС в статті під назвою "Реляція Модель про данні великих банків та загальних даних" (російський переклад першої описаної статті, опублікованій в 1 журналі "СУБД" N 1995). Сьогодні ця модель є справжнім стандартом, заснованим майже на всіх сучасних комерційних базах даних.

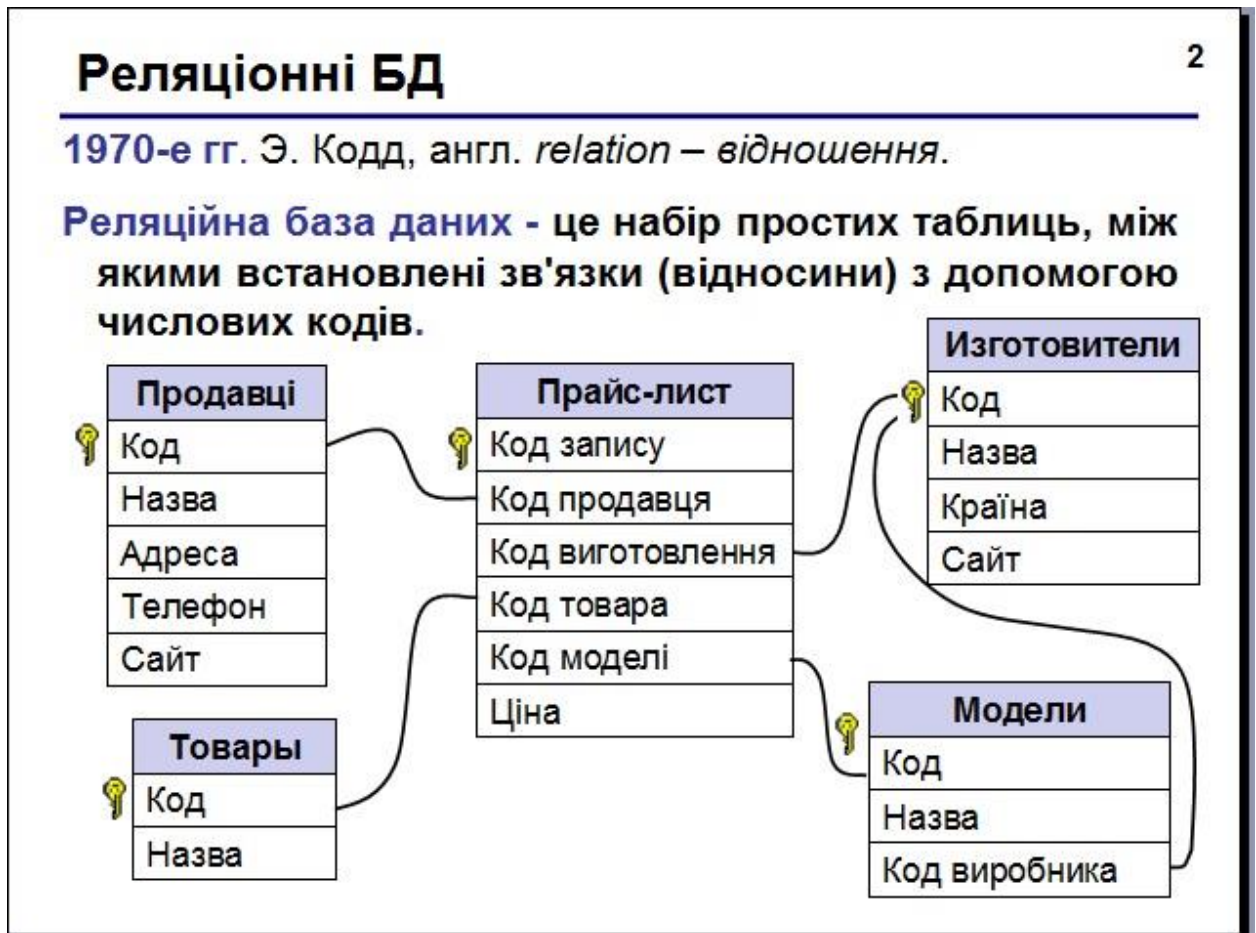


Рисунок 1.5 - Огляд реляційної бази даних

Реляційна модель забезпечує набагато більш високий рівень абсорбції даних, ніж ієрархічна або мережева модель. У статті Кодада йдеться, що "реляційна модель забезпечує інструменти для визначення даних, заснованих виключно на її природній структурі, тобто без необхідності додавання додаткових структур для забезпечення роботи машини." Іншими словами, уявлення даних не залежить від їх фізичної організації. Це досягається за

допомогою математичної теорії відносин (один і той же "реляционное" назва походить від англійських відносин - "відносини").

До моделі даних складу реляційної зазвичай включають теорію нормалізації. Крістофер Дейт (рис.1.6) сформував три складові частини реляційної моделі даних:

- структурна
- маніпуляційна
- цілісна

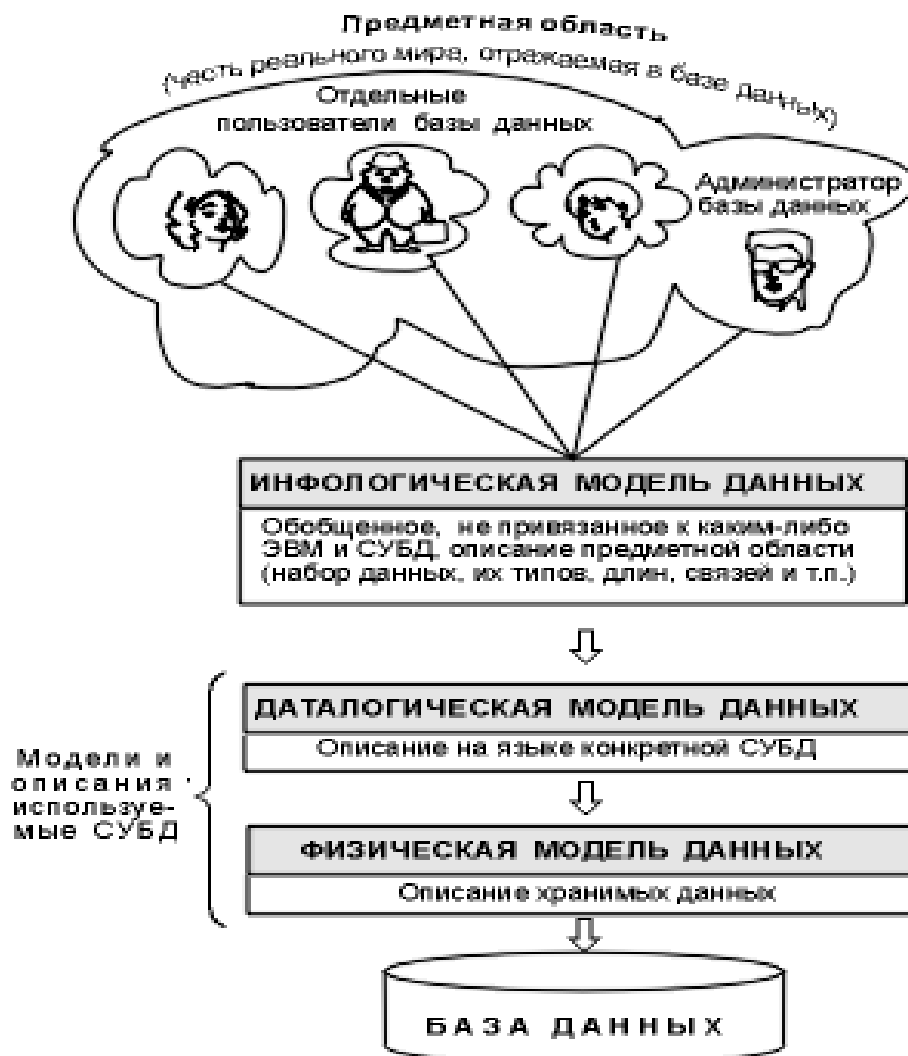


Рисунок 1.6 - Реляційна модель Крістофера Дейта

Структурна частина моделі визначає, що єдина структура даних є стандартизованою n-ary ставленням. Відносини представлені у вигляді

таблиць, де кожен рядок є кортежем, а кожен стовпець - атрибутом, певним в деяких областях. Такий неформальний підхід до концепції взаємин забезпечить розробників і користувачів більш знайомим форматом подання, в результаті чого реляційна база даних стане останньою таблицею.

Маніпулює частина моделі визначає два фундаментальних механізми маніпуляції даними: реляційна алгебра і реляційний бюстгальтер. Основною функцією маніпулює частини реляційної моделі є забезпечення реляційних вимірювань. Релеляційні бази даних будуються з використанням математики, проте вона дуже проста, на рівні алгебри.

Інтегрована частина моделі визначає цілісність об'єкта і вимоги цілісності зв'язку. Перша вимога полягає в тому, що будь-який зв'язок відрізняється від будь-якої іншої діяльності, тобто є ключем до будь-якого зв'язку. Вимога щодо цілісності з'єднання або зовнішнього перемикача полягає в тому, що зовнішній перемикач являє собою або кортеж з одним і тим же значенням первинного перемикача стосовно до кожного значення зовнішнього перемикача, або значення зовнішнього перемикача є нульовим (тобто не вказано).

Можна провести схожість між елементами реляційної моделі даних і елементами моделі зв'язку з суб'єктом. Відносини співвідносяться з сутностями і наборами труб. Таким чином, модель стовпчика "entity-link", яка представляє реляційні відносини в таблиці, називається атрибутами.

Кожен атрибут визначається в домені так, щоб його можна було розглядати як поточне значення атрибута. В одній і тій же області можна визначити кілька характеристик однієї і тієї ж взаємозв'язку і навіть атрибути різних взаємин.

Система управління базами даних - це програмний пакет, що забезпечує централізоване управління базами даних і доступ до даних. СУБД виступає в якості інтерфейсу між користувачами і базою даних.

СУБД забезпечує програмні пов'язані з уведенням-виводом даних, засоби для створення, завантаження, запиту й вирішує питання спільного їхнього використання й захисту відновлення даних, контролює дії.

СУБД забезпечує ефективний доступ користувачів служить для підтримки бази даних в актуальному стані й до даних, що втримуються в ній, у рамках наданих користувачам повноважень.

За ступеню універсальності розрізняють два класи СУБД (рис. 1.7):

1. системи загального призначення;
2. спеціалізовані системи.

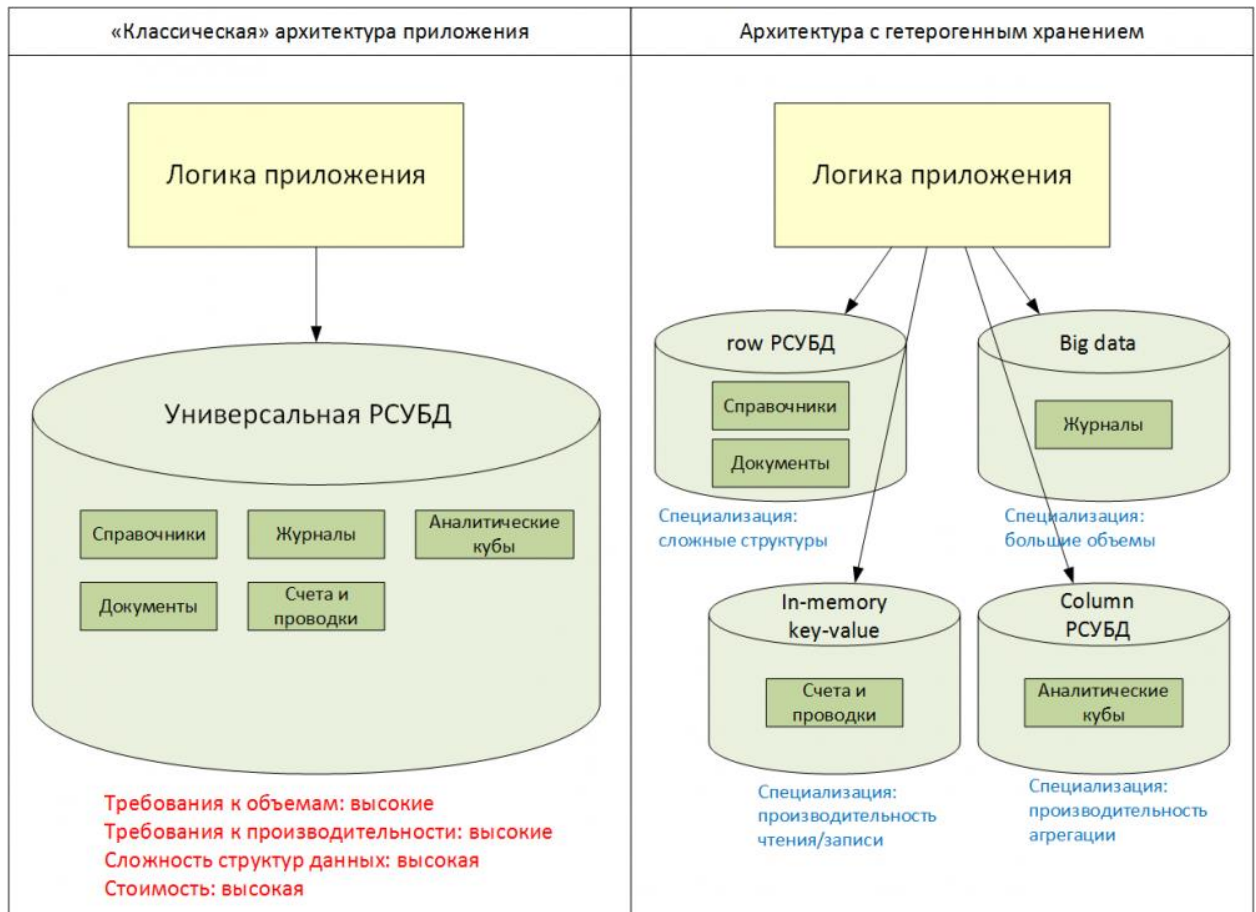


Рисунок 1.7 - Приклад системи загального призначення (1) та спеціалізованої системи (2)

CMS загального призначення не адаптовані до будь-якої теми або інформаційним потребам будь-якої групи користувачів. Кожна система цього

типу реалізується як програмний продукт, розроблений на деяких комп'ютерних моделях в конкретній операційній системі, яка доступна багатьом користувачам в якості комерційного продукту. був налаштований на роботу з конкретною базою даних.

економить робочу силу. Ці бази даних розробили функціональні і навіть функціональні наднаціональні функції. Приватні бази даних створюються рідко, оскільки СУБД не є загальноновизнаними. Загальна мета СУБД полягає у виконанні всіх функцій, пов'язаних з функціонуванням бази даних інформаційної системи

- дозволяє створити структуру бази даних для визначення початку завантаження вихідних даних.

- вони керують можливостями користувача для доступу до бази даних, організовуючи паралельний доступ до бази даних з різними турнірами. - збереження логічної і фізичної цілісності даних

- захист від ліній ведення операцій.

СУБД покладаються один на одного з типів моделей даних, які є мережевими, ієрархічними або реляційними.

СУБД, одна з найбільш важливих позначень СУБД, надає кілька пропозицій СУБД бази даних для цілей, захисту і відновлення даних. Сучасні засоби зв'язку і газонокосарки високого рівня роблять СУБД корисним інструментом для кінцевих користувачів

Особливості структури бази даних (опис):

- інструменти для розробки форм на екрані, онлайнної перевірки, перегляду і обробки

- інструменти для створення зразків даних при певних умовах, а також операції обробки, інструмент для генерації звітів з бази даних для користувачів-дружня друк результатів

- макроси, вбудований алгоритмічну мову (Dbase, VisualBasic або інший), мова запитів (QBE за прикладом, SQL) і т.д. це він використовується

для реалізації процедур обробки подій в призначених для користувача завданнях, а також нестандартних алгоритмів обробки даних,

- інструментів для створення користувацьких додатків (генераторів додатків, заблокованих меню і панелей управління додатками) використовуються для дозволу операцій з базою даних з використанням одного технологічного процесу. Потужність бази даних і бази даних.

Основними характеристиками СУБД і бази даних є:

- дані не можуть дублюватися на різних об'єктах моделі;
- єдиний введення даних і легкий збір даних - несуперечливість даних -

БД цілісність

- множинні засоби доступу, всі типи вибірок даних і їх використання в різних завданнях користувача; аварійні ситуації, збої в роботі апаратного і програмного забезпечення, захист і відновлення даних в разі помилок користувача;

- захист даних від несанкціонованого доступу шляхом дискримінації для різних користувачів

- можливість зміни структури бази даних без повторного блокування даних

- гарантія незалежності програм даних дозволяє реєструвати програму при зміні структури бази даних,

- реорганізація розподілу даних бази даних на машинному носії для поліпшення об'ємних і часових характеристик бази даних

- пошук високорівневого кінцевого користувачаорієнтований мову запитів, що забезпечує видалення інформації з бази даних за всіма видами контенту у вигляді відповідних і зручних для користувача інформаційних форм.

Технологія СУБД є основою для забезпечення доступу користувачів до різних тематичних галузях.

Критерії відбору користувачів для СУБД. Вибір СУБД для призначених для користувача додатків визначається багатьма факторами, такими як

технічне і базове програмне забезпечення, конфігурація, робота і дискова пам'ять вимоги для нових користувальницьких додатків:

- підтримуваний тип моделі даних, предметна область, інформація, топологія і логічна модель
- експлуатаційні вимоги до обробки даних
- функціональність бази даних; рівень досвіду користувачів по СУБД і наявність онлайнових транспортних засобів для поліпшення БД і взаємодії з цією БД системою.

Пара під назвою "Attribute Name - Domain Name" називається схемою. Потужність, привласнена цього набору, називається ступенем відносини або "арифметикою". Іменованій набір схем взаємин є базовою операцією. У нашому випадку, ключ - це "номер послуги", тому що це означає різні для кожного співробітника компанії. Якщо трубки визначені лише кількома ознаками, вони кажуть, що корабель з'єднаний. Зв'язок може містити кілька ключів. Ключ завжди є первинним ключем, його значення не можуть бути поновлені. Всі інвестиції у відносини називаються можливими ключами. Повторювані ключі використовуються для демонстрації зв'язків з різними відносинами. Зміна структури реляційної моделі:

Легко зрозуміти і легко зрозуміти. Сукупної структурою інформації є "таблиця", правила проектування, засновані на математичному пристрої та незалежності даних. При запуску БД додатки зміни є мінімальними; в зовнішньому патчі відсутня конкретна організація БД для редагування запитів і записи БД-запитів додатка. "Таблиці" в результаті логічної схеми, яка містить багато "таблиць". Це ускладнює розуміння структури даних; БД отримує відносно велику зовнішню пам'ять завдяки відносно низькому рівню доступу до даних.

Переваги реляційної моделі:

- простота і зрозумілість для користувача системи. Таблиця використовується як єдина інформаційна конструкція;
- строгі правила проектування, які базуються на математичному апараті;

- повна автономність даних. В програмі зміни мають великого впливу на вміст даних які знаходяться в таблиці.
- зовнішня пам'ять БД не має знати організацію конкретних запитів та записів.
- Недоліки реляційної моделі:
 - не завжди можна уявити предметну область у вигляді таблиці;
 - під час роботи з базою даних з'являється велика кількість таблиць. Такі таблиці не зручно редагувати та керувати;
- БД займає відносно багато зовнішньої пам'яті;
- відносно низька швидкість доступу до даних



Рисунок 1.8 - Приклад реляційної бази даних з вхідною інформацією

У даній дослідницькій роботі також буде зустрічатися поняття «система підтримки прийняття рішень».

Система підтримки прийняття рішень - автоматизована система (програмне забезпечення) надавати допомогу і підтримку різних видів людської діяльності в процесі прийняття рішень по структурованим або неформальним питань. Стагнація СППР забезпечує надійний і об'єктивний аналіз процесу прийняття рішень у важких умовах. [9].

"СППР - це специфічна і чітко визначена система, заснована на персональних комп'ютерах". Так багато визначень систем підтримки

прийняття рішень з широким діапазоном способів, розмірів і типів СППР. На основі діапазону розподілу і використання СППР.

Команда може використовувати систему для своїх власних цілей, а інструмент для менеджера (наприклад, оператора системи) може виконати завдання. Система містить моделі (звані також аналітичними інструментами), які можуть використовуватися для аналізу даних. Дані, які супроводжують або супроводжують аналіз. Процес прийняття рішень підтримується САТР і є досить складним з двох причин.



Рисунок 1.9 - Схема циклу управління системою

Мета або метод його отримання використовується користувачем в поєднанні з іншою інформацією для визначення того, чи є більше інформації,

а не прийняття рішень. Заснована на ПК система підтримки прийняття рішень включає в себе безліч систем, інструментів і технологій.

Система не є процесом, званим онлайнним контрактним процесом, і класифікується. Завдання в області прийняття рішень, підтримувані СППР, є складними з двох причин: або ціль, або інструменти для досягнення цієї мети неясні.

СППР не приймає рішень, але надає інформацію, яка використовується користувачем для прийняття рішень з іншою інформацією.

Заснована на ПК система підтримки прийняття рішень включає в себе широкий спектр систем, інструментів і технологій. Комп'ютеризована система класифікується як онлайнним процесом оброблення транзакцій, якщо вона не називається процесом онлайнної обробки.

Якщо програма встановлена на персональний комп'ютер і може допомогти адміністраторам приймати рішення, вона також буде належати SPD. Адміністративні інформаційні системи (БМР), географічні інформаційні системи, інтерактивні операційні системи (ОПЛАП), програмне забезпечення для картографування інформації і системи підтримки групових рішень можуть бути віднесені до категорії систем СППР.

Для досягнення управлінських цілей необхідно звернути увагу на два головні типи СППР. Виділяють СППР:

1. корпоративні (Enterprise-wide) або широкі, підприємницький рівень;
2. ручні комп'ютери (desktop) СППР.

Торговці підприємств, підключення до великих сховищ даних, призначені для багатьох менеджерів в різних компаніях і призначені для одного користувача і використовують невеликі системи на персональних комп'ютерах менеджера. Зосередження уваги на прийнятті обґрунтованих і взаємовигідних рішень на корпоративному рівні призвело до розгортання ГСМР з комплексними сховищами даних, які дозволяють особам, які приймають рішення, знаходити майже всю інформацію про вашу компанію в потрібний час.

Для виконання ОПР використовують корпоративні СППР операції:

- детальне оброблення інформації (drill-down),
- окремі зрізи або розділи (slice і dice),
- наочно (графічно і схематично) зображення підприємства та дані зовні.

Підприємство СППР спрощує процес прийняття рішень від дуже простих інтегрованих інформаційних систем до аналітичних передових систем адміністративної інформації.

Ідентифікація ССОО підприємства:

1. Системи доступу до елементів даних.
2. Системи аналізу інформації, що полегшують обробку даних для пошуку прихованих зразків інформації з використанням комп'ютеризованих аналітичних засобів, таких, як пакети статистичних даних, засоби отримання інформації (видобуток даних

3. Аналітичні системи, заздалегідь певні моделі та діаграми і компоненти, що забезпечують доступ до баз даних або вікнам даних, заснованим на рішеннях, забезпечують підключення до сховища бізнес-даних.

Найбільш відоме підприємство СППР доповнює і розширює Manager Information Systems, які були розширені в 1980 році в ХХ. Використовує сучасну графіку, комунікаційні лінії, і методи зберігання, що забезпечують адміністраторам легкий доступ до інформації про стан справ в компанії.

Організації часто встановлюють тісний зв'язок між корпоративним СППР, сховищами даних і настільним СППР. Де знаходиться інформація, як вона аналізується і подається.

Організація має розробити структуру СППР для власного використання. Структура СППР може бути націленою:

- а) на підприємство в цілому;
- б) користувач окремо від підприємства.

Орієнтована на клієнта (Загальнокорпоративна) структура може мати канал зв'язку для передачі та аналізу інформації з настільного комп'ютера користувача, інструментів СППР, на сервер і серверні інструменти СППР. САП і сховища даних можна знайти в будь-якій точці організації.

Користувач СППР не отримав стільки запитів від одного користувача, скільки від корпоративного СППР. Іноді електронні таблиці, такі як Excel або Lotus1-2-3, використовуються для виконання необхідного аналізу робочого столу або для розробки призначених для користувача функцій СППР для адміністраторів, щоб створити окремі інструменти для підтримки користувальницьких рішень.

Всі види САП дозволяють керівникам відповідати на питання, пов'язані з прийняттям рішень.

1.2 Аналіз інформаційно-аналітичних систем та систем підтримки прийняття рішень, використовуваних у сфері кібербезпеки

Збираючи матеріали до написання роботи про інформаційно-аналітичні системи (зокрема, системи підтримки прийняття рішень), пов'язані з кібербезпекою, виявилось що відбувалось не мало україномовних досліджень по цій темі.

Так, наприклад, у своїй роботі «Метод імітаційного моделювання функціонування СППР в складі програми інформаційної безпеки» Зибін С. В. пропонує математичну модель СППР, яка в теорії може бути застосована для побудови СППР, пов'язаної з інформаційною безпекою: «синтез системи технічного захисту і підтримки прийняття рішень для них, які будуються на основі засобів обчислювальної техніки повинен здійснюватися відповідно до наступних критеріїв:

- декомпозиція процесу керування, тобто можливість реалізації будь-якої складної операції на послідовності більш простих;
- модульність побудови систем;
- магістральний спосіб обміну інформацією, який дозволяє мінімізувати кількість зав'язків;
- можливість масштабування обчислювальної потужності за рахунок використання розподілених систем» [9][???].

Після опису, без сумнівів, великого плану йде доволі сухий опис математичної імітаційної моделі, яку автор пропонує використати за основу створення СППР для захисту інформації. Варто наголосити про те, що в статті не були запропоновані критерії вибору ЗЗІ, а самому захисту інформації увага у статті майже не приділяється.

Враховуючи об'єм задуму та високу модульність результату, кількість витрачених годин на розробку подібної уніфікованої системи керування та інтегрування в неї ЗЗІ може сягнути аж надто високих значень. Враховуючи постійні оновлення ЗЗІ та відкидання деякими розробниками підтримки старих версій свого програмного забезпечення, підтримувати таку систему буде вкрай важко.

Обґрунтованою є і стаття Дрейса Ю. О. та Ковалю Л. В. «Система підтримки прийняття рішень процесу управління захисту інформації в інформаційно-телекомунікаційній системі». С самого початку статті наведено, що було «проведено розробку системи підтримки прийняття рішень процесу управління захисту інформації (ЗІ) в інформаційно-телекомунікаційній системі на основі експертного оцінювання використаних методів та засобів ЗІ відносно деякого часу для надання рекомендацій щодо контролю, управління та підвищення ефективності цього процесу» [10][]. Надалі у роботі йде докладний опис, з використанням математичних моделей та знімками практично реалізованого ПЗ, побудованої ними СППР, що

враховувала цілий ряд показників, наприклад наявність та ефективність концепції безпеки інформаційно-телекомунікаційної системи або служби захисту інформації, та в кінці кінців після наповнення експертом цих даних протягом деякого періоду часу надавала рекомендації щодо поліпшення конкретних заходів безпеки заради покращення безпеки в цілому.

Проте, в цій роботі КСЗІ використовується лише як один із безлічі факторів, а окремі компоненти КСЗІ не беруться до уваги, як елемент єдиної КСЗІ.

Праця «Безпека систем підтримки прийняття рішень» без сумніву стосується як безпеки, так і СППР – натомість фокусує на забезпечення безпеки для СППР. Як результат, зміст цієї роботи є наголошення на наступному: «Питання забезпечення кібербезпеки СППР, як витікає з попередніх розділів, у загальному уявленні фактично зводиться до запобігання несанкціонованого доступу (unauthorized access to information) до інформації і до сервісів системи.»

САТД являє собою комп'ютеризовану інформаційну систему, яка використовується для підтримки різних видів діяльності по прийняттю рішень в тих випадках, коли автоматизована система, в повній мірі забезпечує весь процес прийняття рішень, неможлива або краща. Користувач безумовно є адміністратором або фахівцем з персоналу (наприклад, фахівець з фінансового планування). Працівник є постачальником послуг для своїх власних цілей або для адміністратора (наприклад, системний оператор). Програмний інтерфейс застосовується до таких систем як діалогові вікна, що полегшує використання системи. Система містить моделі, використовувани для аналізу даних (також звані аналітичними допоміжними засобами). Дані, які супроводжують або підкріплюють аналіз. Конфігурація ВСПР не визначена, однак вона забезпечує інформацію, використовувану користувачем для того, щоб зосередитися на прийнятті рішень на

Загальноорганізаційна рівні з використанням голосового зв'язку і взаємопов'язаних управлінських рішень, що веде до створення загальноорганізаційних САТ зі всеосяжним сховищем даних, що дозволяє директивним органам (ІНР) або особам, які приймають рішення, знайти майже всю інформацію про їхній бізнес точними

момент. Тому учень повинен зрозуміти, що система підтримки прийняття рішень є комп'ютеризованою системою, яка взаємодіє з іншими системами для надання допомоги керівникам в ухваленні рішень. Системи підтримки прийняття рішень (СОР) - це інформаційні системи, основна мета яких полягає в забезпеченні комп'ютерної підтримки для вирішення погано структурованих і неструктурованих питань управління бізнесом на різних етапах прийняття рішень і аудиту. Незважаючи на те, що в усьому світі розроблені сотні видів СППР, ці системи в Україні практично відсутні. Україні практично не використовуються.

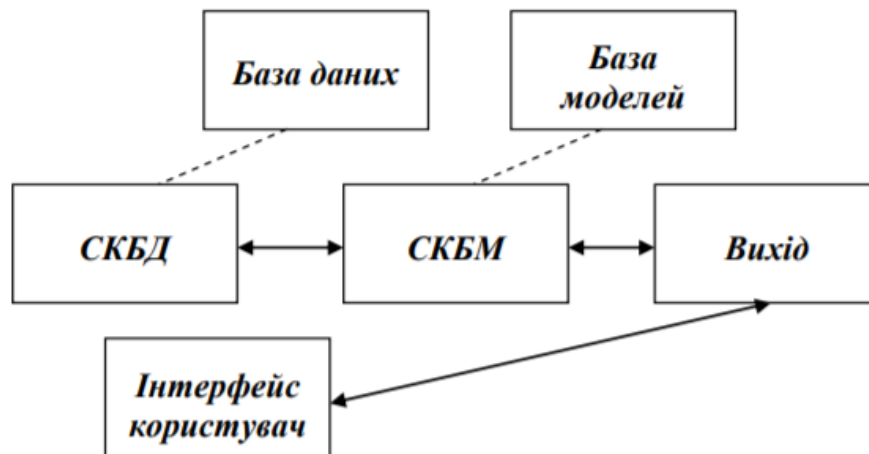


Рисунок 1.10 Класична структура

Компоненти СППР отражают официальное функциональное разделение системы на подсистемы, отличающиеся от конструкции системы, то есть в первую очередь создает базу данных СУБД Exit Database User Interface, систему управления данными и систему управления моделями; используя концепцию своей архитектуры.

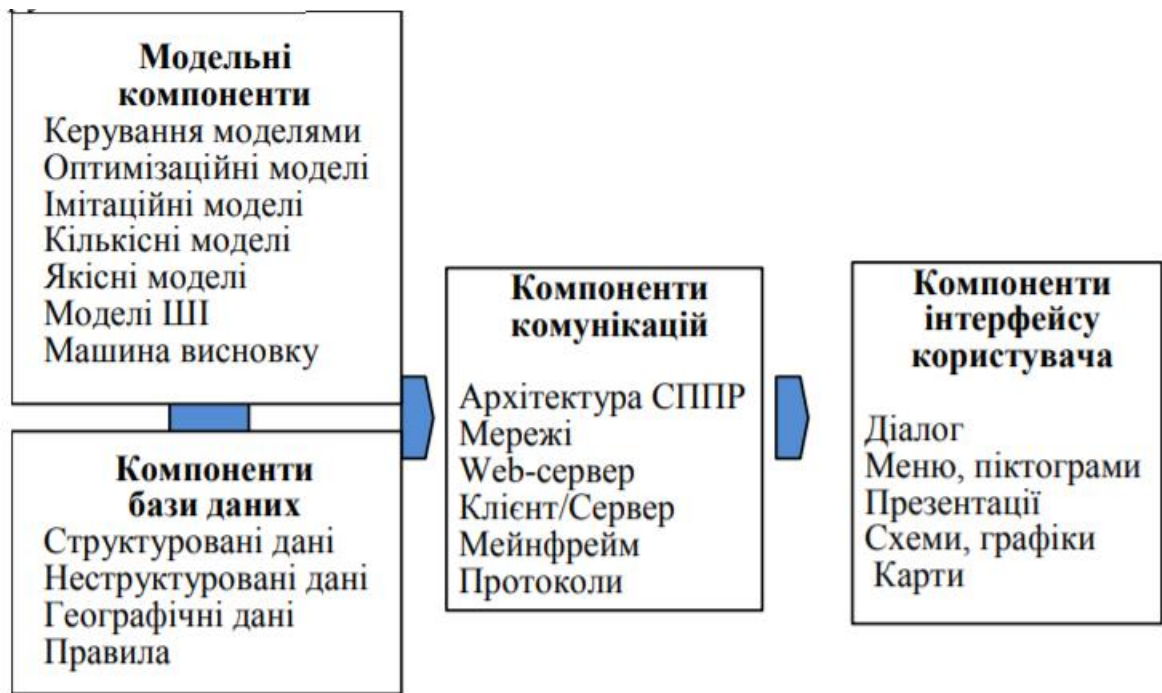


Рисунок 1.11 Компоненти підсистем СППР

СППР може містити різні типи моделей. Наприклад, статистичні моделі: Регресійний аналіз, аналіз змінних і експоненціальне розподіл. Моделі обліку включають цінові моделі, бюджети, податкові плани і аналіз витрат. Модель управління персоналом, наприклад, міністерство праці. Маркетингові моделі включають в себе стратегічний аналіз реклами та модель відбору споживачів. Характеристики всіх цих моделей варіюються в залежності від ступеня їх використання. Кожен з них являє собою конкретне спрощення процесу прийняття рішень, що корисно для розуміння фактів. Це вимагає здатності створювати і використовувати ці моделі таким чином, щоб користувачі з не великим досвідом могли її легко використовувати. Основа моделі СППР містить моделі, які не оптимізовані або не узяті в вибірку. Моделі оптимізації включають математичні моделі програмування - лінійні (розподіл ресурсів, оптимальне планування, графічний мережевий аналіз, транспортна проблема), нелінійні динамічні моделі; моделі обліку; моделі аналізу вартості для визначення інвестиційної стратегії; моделювання маркетингу і т.д. Статистичні моделі не є оптимальними

(Лінійний і нелінійний регресійний аналіз) методи оцінки часових рядів (аналіз); альтернативні методи моделювання (наприклад, моделювання) і т.д. Система управління моделями є компонентом універсальної архітектури СППР. Функції системи полягають в забезпеченні доступу до класифікації, редагування і моделей, аналогічним функцій системи управління базами даних. Мал. Діаграма управління 6.3 моделей запропонована з важливими елементами: Основа моделі, управління моделлю і управління діалоговим вікном в якості основної частини інтерфейсу користувача. Основними функціями SWSC є: створення нових моделей; каталогізація широкого спектру моделей; комбінування компонентів моделей і бази моделей; виконання різних загальних управлінських функцій. Заснована на моделі система керування СППР пропонує користувачеві цілий ряд можливостей, включаючи легкий доступ до моделей, допомагає зрозуміти результати моделювання, інтегрує моделі і дозволяє досліджувати точність рішень, надає інструменти для управління моделями, і дозволяє впроваджувати зовнішні моделі системи які використовує підприємство в своїх цілях.

Інші публікації, такі як «Моделювання систем захисту інформації» Антонюка А. О., занурюються в моделювання систем забезпечення безпеки всередині існуючої інформаційної системи, а не організації, і такі дослідження не можуть бути використаними для створення ІАС та СППР, призначених для вибору необхідних ЗЗІ.

Підводячи підсумки, маємо бачення, що в українських публікаціях інформаційна безпека, ІАС та СППР є досліджуваними, проте не з боку ІАС та/або СППР для вибору засобів захисту для організації. Особливо, фокус у цих роботах на КСЗІ, як правило, відсутній.

Табл. 1.2.Класифікація СППР

Класи класифікації	Суть (основа) визначення	Групи за кваліфікацією (види систем)
Модель концептуальності	Підхід інформаційний	Модель концепції Модель еволюція СППР
	Використання знань	Орієнтовані на правила та знання СППР
	Підхід інструментальний	Спеціальні (робочі) СППР СППР-преобразувачи СППР-інструменти
Користувачі системи	Управління шляхом єрархії	Термінальний режим Режим клерка Режим посередника Автоматизований режим
	Спосіб взаємодії користувача з системою	Особиста підтримка Групова підтримка (групові СППР) Підтримка організації (багатокористувацькі, інтер-організаційні, інтра-організаційні СППР)
	Ступінь залежності осіб у процесі прийняття рішення	Вища ступінь керування (виконавчі інформаційні системи) Проміжня ланка керування Низова ланка керування
Побудова завдання по прийняттю рішень	Актуальність завдання	Особливі проблеми спричиненні в даному випадку(СППР на даний випадок)Проблеми які повторюються(циклічні) проблеми якінституції прийняття рішення

Продовження табл. 1.2.

	Опис обраної проблеми	Вибір за повнотою вибору Багатокритеріальний вибір (наприклад, СППР Decision Grid)
	Вид моделі	Об'єктивна модель Суб'єктивна модель
	Підтримка функцій	Функціонально-специфічні СППР СППР загального призначення
Засоби забезпечення	Прийняття рішення	СППР, данні як ціль СППР, модель як ціль СППР, курс на документи СППР, курсівані на комунікації Веб ставка СППР
	Ступень мов інтерфейсу користувача	Мова процедурна, мова командна, мова непроцедурна, природна мова
Застосування у галузі	Сфера професії	Мікроекономіка, Макроекономіка Конторська діяльність (офісні СППР) Таксакція розповсюдження технологій Юриспруденція Медицина і т. ін.
	Горизонт часу	Стратегічне керування (довгострокові рішення) Тактичне керування (середньострокові рішення) Операційне керування

У жодній із праць не було згадано офіційних переліків ЗЗІ, дозволених для захисту державних інформаційних ресурсів, на чому варто наголосити.

1.3 Аналіз поточного стану вибору засобів захисту інформації

Комплекси систем захисту інформації, які будуються для захисту інформації, та містять державні інформаційні ресурси, зобов'язані використовувати лише ті засоби, що визначені нормативними документами – а саме, «Переліком засобів технічного захисту інформації, дозволених для забезпечення технічного захисту державних інформаційних ресурсів та інформації, вимога щодо захисту якої встановлена законом» [12] та «Переліком засобів КЗІ, які мають експертний висновок за результатами державної експертизи у галузі КЗІ» [13]. Інші види засобів захисту заборонені для використання при побудові комплексної системи захисту інформації. З цього витікає необхідність в постійному звірванні із вищезазначеними переліками. Цей перелік включає терміни інструментів ТЗІ (обладнання, основна функція якого полягає в захисті інформації від витоків, перебоїв і загроз блокування; обладнання з основною функцією плюс функції інформаційної безпеки; і обладнання, спеціально розроблене або розроблене для ідентифікації іпотечних кредитів і представляє загрозу для інформації; Транспортні засоби, спеціально сконструйовані або пристосовані для оцінки інформаційної безпеки), випробовуються на основі свідчення про відповідність або позитивного експертного висновку відповідно до процедури, встановленої законом; чи відповідає нормативна документація ТЗІ вимогам нормативної документації ТЗІ: Правилами проведення робіт із сертифікації засобів захисту інформації, затвердженими спільним наказом Адміністрації Держспецзв'язку та Держспоживстандарту України від 25.04.2007 р. № 75/91 і зареєстрованими в Міністерстві юстиції України 14.05.2007 р. за № 498/13765, та Положенням про державну експертизу в сфері технічного захисту інформації, затвердженим наказом Адміністрації Держспецзв'язку України від 16.05.2007 р. № 93 і зареєстрованим в Міністерстві юстиції України 16.07.2007 р. за № 820/14087.

1. Розмір таблиці та складнощі пошуку в ній збільшують час на створення необхідного комплексу систем захисту інформації.
2. Відсутність рекомендованих цін на продукцію сповільнює остаточне затвердження бюджету КСЗІ. Будь-яке несхвалення у проекті КСЗІ вертає сторону, що створює КСЗІ, до проблем, описаних у пункті 1, і затрачений час зростає геометрично.
3. Відсутність рекомендованих цін у таблиці також спричиняє можливості для створення корупційних схем з боку сторони, що створює КСЗІ, а саме штучне викривлення цін на продукцію у звітах начальству та подальша закупівля продукції за завищеними цінами. Це призводить до перевитрат бюджету та, можливо, придбання не найкращих альтернатив.
4. Дані переліки періодично оновлюються, що з урахуванням їх обсягу може призвести до помилок у виборі певних засобів, з урахуванням використання застарілої інформації, або не врахування оновлених даних у зв'язку з великими об'ємами даних переліків. [14]

1.4 Актуальність створення бази даних та аналітичної системи вибору

Вищезазначені переліки засобів захисту інформації мають ряд проблем, а саме:

1. Перелік дозволених засобів розміщується в різних офіційних документах.
2. Тільки в одному з документів міститься більше 300 найменувань. При цьому ДССЗЗІ час від часу ці списки оновлюються. Відслідковувати та мати актуальну інформацію стає все складніше.

3. Інформація в таблицях розміщується так, що для експерта з побудови КСЗІ важко розібратися та отримати доступ до бажаної інформації.
4. Класифікація описана важкою для розуміння мовою. Зрозуміла для людини яка має досвід та експертизу в читанні документів НД ТЗІ.
5. Є розмиті назви класифікаторів засобів захисту, наприклад «Програмний комплекс захисту інформації», «Програмне забезпечення» тощо.
6. Також, існують класифікатори, що копіюють один одного, наприклад «Комплекс засобів захисту комп'ютерної програми» та «Комплекс засобів захисту програмного продукту».

Магістерська дипломна робота покликана допомогти експертам в сфері інформаційної безпеки з прийняттям рішення про засоби захисту інформації, які він отримає використовуючи ІАС, яка автоматично буде підтримувати актуальність інформації.

У майбутньому планується створення повністю функціональної інформаційно-аналітичної системи вибору та аналізу усіх дозволених засобів захисту, що полегшить експертам приймати рішення.

Висновок до першого розділу

У даному розділі були висвітлені ІАС та СППР, які використовуються в сфері інформаційної безпеки та кібербезпеки, та нинішній стан вибору засобів ТЗІ для створення комплексної системи захисту інформації.

Системи які використовуються у сфері кібербезпеки, такі як ІАС та СППР мають безліч описів в україномовних наукових працях в більшій мірі теоретичні, не маючи жодного технічного підґрунтя. Всі ці праці не ставлять до уваги переліки дозволених засобів ЗЗІ для захисту державних інформаційних ресурсів.

Наразі для побудови КСЗІ використовується ручна робота з пошуку дозволених засобів ЗЗІ яка розміщується на офіційному сайті ДССЗЗІ. Таким чином експертне рішення та час для його прийняття збільшується.

Жодна робота не ставила для себе ціль спростити життя експерту та економити його час на пошук інформації.

Проблеми які згадані вище, спонукали для проведення аналізу, збору даних та створення інформаційно-аналітичної система вибору та аналізу засобів ЗЗІ, дозволених для використання захисту інформаційних ресурсів держави. Ця робота має полегшити рутині завдання експерта з побудови КСЗІ та повністю автоматизувати етап пошуку потрібних засобі ЗЗІ.

РОЗДІЛ 2

РОЗРОБКА СТРУКТУРИ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ ТА ОКРЕСЛЕННЯ ЇЇ МАЙБУТНЬОГО ФУНКЦІОНАЛУ

Мета цієї роботи полягає у розробці інформаційно-аналітичної системи, яка враховуючі певні запити користувача та спеціальні вимоги, щодо побудови КСЗІ, задані спеціалістом з інформаційної безпеки, а також дані про підприємство буде формувати певні рекомендації. Експерт отримавши ці дані та проаналізувавши їх зможе прийняти рішення з приводу використання певних технічних засобів інформації для побудови КСЗІ.

2.1 Огляд функціоналу інформаційно-аналітичної системи

Інформаційно-аналітично система має базуватися на певних функціях і параметрах які класифікують її як інформаційно-аналітичну систему. До таких параметрів та функцій можна віднести[15]:

1. Механізми, спрямовані на аналіз отриманих вхідних даних;
2. База даних інформаційних відомостей, які беруться до обробки;
3. Завдання поставлене перед інформаційно-аналітичною системою вирішуються шляхом дотримання певних правил обробки інформації;
4. Для того щоб користувач мав змогу взаємодіяти з ІАС, використовується програмно-технічний комплекс;
5. Варіанти рекомендацій та пропозиції відображений модульним функціоналом даних;

Розкриємо вище згадані пункти використовуючи завдання поставлене в цій роботі:

1. Аналітична система побудована таким чином, що експерт не має змоги отримати доступ до перегляду алгоритмів та способів обробки даних. Такий функціонал не передбачено в даній системі. На думку автора, ця інформація не є доцільною та не має жодної цінності для експерта. Експерт отримує сформовані дані, та вже кінцевий результат роботи програми без проміжних етапів. Сформована інформація буде подана експерту для ознайомлення та прийняття рішення без зайвих додатків.
2. Дані зібрані в офіційних джерелах ДССЗІ які знаходяться на сайті <https://cip.gov.ua/> та на основі цих даних адміністратор побудує базу даних для використання системою;
3. Система буде приймати рішення, базуючись на аналізі даних та критеріях відбору, які будуть остаточно обрані в ході експлуатації системи спеціалістом з інформаційної безпеки.
4. ІАС будемо побудована таким чином, що дані які оброблятиме та зберігатиме система будуть надходити в неї з двох окремих сервісів, які знаходяться в середні програми. Перший сервіс – це база даних яка буде під контролем адміністратора баз даних. Він буде тримати її в актуальному стані. Також в системі буде сервіс, до якого буде доступ самого користувача ІАС. Користувач, зможе наповнити систему даними про своє підприємство. Наприклад, яке програмне забезпечення використовується на підприємстві, який клас АС та кількість АС та інше.
5. В рамках цієї роботи буде розроблений прототип інформаційно-аналітичної системи. Цей прототип буде використаний як тестова середа для збору даних про роботу системи з подальшим покращенням та створенням повноцінної ІАС.

На виході, після повної роботи програми, результатом якої будуть сформований перелік рекомендацій засобів ТЗІ для побудови КСЗІ. Ці рекомендації скрадатимуться з певних компонентів. Ці компоненти будуть вміщати в себе не тільки дозволенні засоби ТЗІ, а й інформація про окреме підприємство та його діяльність. Під час формування цих компонентів буде звертатися увага на:

- бажаний бюджет підприємства;
- ризики які підприємство може прийняти;
- побажання з приводу певних засобів захисту інформації;
- побажання з приводу країни постачальника ЗЗІ або рівень гарантії;
- аналіз моделі загроз;
- інформація про інформаційні та автоматизовані системи, які використовуються на підприємстві(згідно з «Тимчасовим положенням про категорювання об'єктів», також відомим як ТПКО-95) програмні продукти, операційні системи та інше.

Для створення інформаційної системи необхідно об'єднати такі компоненти:

Люди: Народи є найбільш важливою частиною інформаційної системи, оскільки без них система не може функціонувати правильно.

Апаратне забезпечення: це частина фізичного компонента інформаційної системи, до якої ми можемо торкатися. Апаратне забезпечення інформаційної системи включає комп'ютер, процесори, монітори, принтер, клавіатуру, диски, iPads, флеш-накопичувачі і т.д.

Програмне забезпечення: Це набір інструкцій, які вказують апаратного забезпечення, що робити. Він може використовуватися для організації, обробки та аналізу даних в інформаційній системі.

Дані: Дані - це добірка фактів. Інформаційні системи працюють з даними. Ці дані можуть агрегуватися, індексуватися і об'єднуватися в таблиці і файли

для формування бази даних. Ці бази даних можуть стати потужним інструментом для кожної системи ділової інформації.

Мережа: вона включає Інтернет, інтранет, екстранет для забезпечення успішної роботи всіх типів організацій і комп'ютерних інформаційних систем.

Процедури: В ньому визначаються принципи, що регулюють функціонування інформаційної системи. У ньому описується конкретний метод обробки і аналізу даних для отримання відповідей, для яких призначена інформаційна система.

Зворотній зв'язок: це компонент інформаційної системи, який визначає, що ІС може бути забезпечений зворотним зв'язком.

Інформаційні системи можна поділити на чотири типи, а саме. Виконавчі інформаційні системи – це стратегічна інформаційна система, яка знаходиться на вершині піраміди. Його головна мета полягає в наданні інформації, зібраної як з внутрішніх, так і з зовнішніх джерел, старшим керівникам та керівництву для аналізу умов, в яких працює організація, і планування відповідних напрямків діяльності для виявлення довгострокових тенденцій. Вона може також використовуватися для контролю за діяльністю організації, а також для виявлення можливостей і проблем. Система ЕІС розроблена таким чином, що вони можуть управлятися безпосередньо керівниками без необхідності залучення посередників.

Виконавчі інформаційні системи виконують такі функції:

- Вона піклується про зручність використання.
- Вона підтримує неструктуровані рішення.
- Вона займається прогнозуванням майбутнього.
- Він відрізняється високою гнучкістю.
- Він є ефективним.

У ньому використовуються як внутрішні, так і зовнішні джерела даних. Він використовується тільки на більш високих рівнях влади. Системи підтримки прийняття рішень DSS або система підтримки прийняття рішень - це комп'ютерна програма, яка використовується старшими менеджерами для аналізу бізнес-даних і подання їх в тій формі, в якій користувачі можуть легше приймати бізнес-рішення. Ці системи зазвичай є інтерактивними і можуть використовуватися для вирішення погано структурованих проблем в організації. Вона допомагає обмінюватися інформацією всередині організації.

Система підтримки прийняття рішень виконує наступні функції:

- Вона підтримує погано структуровані або напівструктуровані рішення.
- Він використовується старшими керівниками.
- Він має аналітичний та / або модельним потенціалом.
- Вона займається прогнозуванням майбутнього.

Система управлінської інформації - це використання інформаційних технологій, людей і робочих процесів для реєстрації, зберігання, маніпулювання і обробки даних для отримання значимої інформації. Ця інформація допомагає особам, які приймають рішення, приймати правильні і точні рішення. Він використовується для прийняття тактичних рішень (середньострокових рішень) для забезпечення безперебійного функціонування організації. Вона також допомагає оцінювати ефективність роботи організації шляхом зіставлення попередніх заходів з поточними заходами. Роль систем управлінської інформації полягає в наступному:

- Вона заснована на внутрішніх інформаційних потоках.
- Вона підтримує щодо структуровані рішення.
- Вона є негнучкою і має невеликий аналітичний потенціал.
- Він використовується керівниками нижчої та середньої ланки.
- Вона стосується минулого і являє, а не майбутнього.

Система обробки транзакцій є тип системи обробки інформації для ділових операцій, пов'язаних зі збором, зберіганням, модифікацією і пошуком всіх транзакцій даних підприємства. Характеристики системи обробки операцій включають надійність, ефективність і узгодженість. ТРС також відомий як обробка в реальному часі.

Система обробки операцій виконує такі функції:

- Вона готує інформацію для інших систем.
- Він використовується оперативними співробітниками і керівниками.
- Вона орієнтована на ефективність.

Розробка інформаційних систем - це комплекс заходів, методів, передових методів, результатів і автоматизованих інструментів, які кожна організація використовує для розробки і постійного вдосконалення інформаційних систем і пов'язаного з ними програмного забезпечення.

Для розробки інформаційної системи можна використовувати чотири етапи, а саме:

1. Визначення та розуміння проблем. Мета першого кроку полягає в тому, щоб визначити масштаби проблеми і знайти рішення. На цьому етапі враховуються також ресурси, час, витрати та інші потреби інформаційної системи.
2. Розробка альтернативного рішення. Мета цих кроків - знайти шлях до вирішення, визначеним системним аналізом. На цьому етапі деякі рішення вимагають модифікації існуючої системи, деякі рішення не вимагають інформаційної системи, а деякі рішення вимагають нової системи.
3. Оцінка і вибір найкращого рішення. Мета третього етапу полягає в оцінці питань здійсненності, пов'язаних з фінансовими, технічними та організаційними аспектами. Він вимірює час і вартість розробки інформаційної системи. Вона оцінює комерційну цінність системи і знаходить оптимальне рішення для розробки інформаційної системи.

4. Впровадження даного рішення. Мета останнього кроку полягає в розробці докладної специфікації конструкції інформаційної системи. Цей етап забезпечує повну реалізацію для вибору і придбання апаратних засобів.

Розробка і програмування програмного забезпечення:

- Випробування, наприклад випробування блоку, системи, приймальні випробування
- Навчання і документація (онлайнова практика, покрокове навчання)
- Перетворення, тобто перехід від старої до нової системи
- Виробництво і технічне обслуговування (огляд, об'єкти, модифікація)

Розуміючи те, що одночасне врахування всіх бажаних компонентів як: бюджет, якість захисту, обрання певних постачальників та особисті побажання – не можливе, саме тому буде створений певний перелік конфігурацій, який має на меті допомогти полегшення з обранням компонентів. Такі рекомендації як:

- Побудова конфігурації з урахуванням АС, бюджету підприємства та ризиків які підприємство може прийняти;
- Побудова конфігурації з урахуванням АС, бюджету підприємства та аналізу моделі загроз;
- Побудова конфігурації з урахуванням АС, експертної оцінки та моделі загроз;
- Побудова конфігурації з урахуванням АС враховуючі всі компоненти.

Використовуючи рекомендації, з приводу обрання компонентів, дозволить більш гнучко використовувати систему експерту. Буде надано змога отримати модель більш приближену до реальних умов та втілення з урахуванням можливостей підприємства. Отримання певної “золотої середини”.

2.2 Структура інформаційно-аналітичної системи

В своїй роботі, Д. Кожевніков[16] розповідає про те, що Інформаційно-аналітична система повинна складатись з трьох рівнів збору інформації, які визначають суть структури і демонструють динаміку стану складних технічних систем: 1) базової інформації про систему інженерних комунікацій, побудовану на основі паспортів об'єктів; 2) нормативно-технічної інформації, який регламентує процеси використання системи інженерних комунікацій, порядок проведення робіт з технічного випробування, оцінки результатів технічного випробування, аналізу ризиків використання об'єктів інженерних комунікацій; 3) оперативної інформації, отриманої при проведенні технічної перевірки. Таким чином виділяємо три основні структури:

- Інформація на вхід;
- Блок проведення розрахунків;
- Побудова оцінки і прогнозів.

За результатами виконання першого етапу комплексного аналізу формується інформаційна модель даних про інженерні комунікації, об'єкти його інфраструктури та елементах прилеглої місцевості, а також створюється розділ бази даних інформаційноаналітичної системи, який може містити необхідні відомості з проектною, виконавчої та експлуатаційної документації, результати виконаних раніше діагностичних обстежень і випробувань, результати топографо-геодезичних робіт та інженерних вишукувань і ін. Другий етап комплексного аналізу полягає в аналізі та співставленні результатів діагностичних обстежень і включає в себе: аналіз результатів внутрішньотрубної дефектоскопії (оцінка дефектів труб і динаміки зміни втрат металу труби при наявності дефектів, оцінка працездатності ділянок інженерні нерного комунікацій з урахуванням зростання розмірів корозійних дефектів, ранжування дефектів за ступенем небезпеки); аналіз результатів обстежень стану захисного покриття і засобів

електрохімізахисту; аналіз результатів технічного діагностування переходів через автомобільні дороги та залізничні дороги та інших структурних елементів інженерних комунікацій. Результатом роботи є перелік потенційно небезпечних ділянок інженерних комунікацій, пропозиції щодо проведення додаткових діагностичних і розрахунково експериментальних робіт, а також пропозиції щодо проведення ремонтних робіт на ділянках.

В результаті другого етапу комплексного аналізу буде визначено вся інформація, необхідна для отримання оцінок ризику експлуатації ділянок інженерних комунікацій. Сформована на першому етапі комплексного аналізу інформаційна модель інженерних комунікацій буде доповнена інформацією, отриманою при проведенні другого етапу комплексного аналізу. На основі сформованої на попередніх етапах інформаційної моделі інженерних комунікацій можна обґрунтувати й апробувати структуру інформаційноаналітичної системи (ІАС) для аналізу ризику експлуатації інженерних комунікацій за результатами обстежень, і як наслідок, попередження природнотехногенних аварій.

Базуючись на даних отриманих з переліку функціоналу ІАС вище, складемо структурно-логічну схему (рис.2.1) з блоком інформації на вхід, блоку проведення розрахунків та оцінки і побудови прогнозів. Прогнози які будуть використані в під час побудови системи для пошуку та аналізу даних для використання експертом з інформаційної безпеки. В результаті другого етапу комплексного аналізу буде визначено вся інформація, необхідна для отримання оцінок ризику експлуатації ділянок інженерних комунікацій

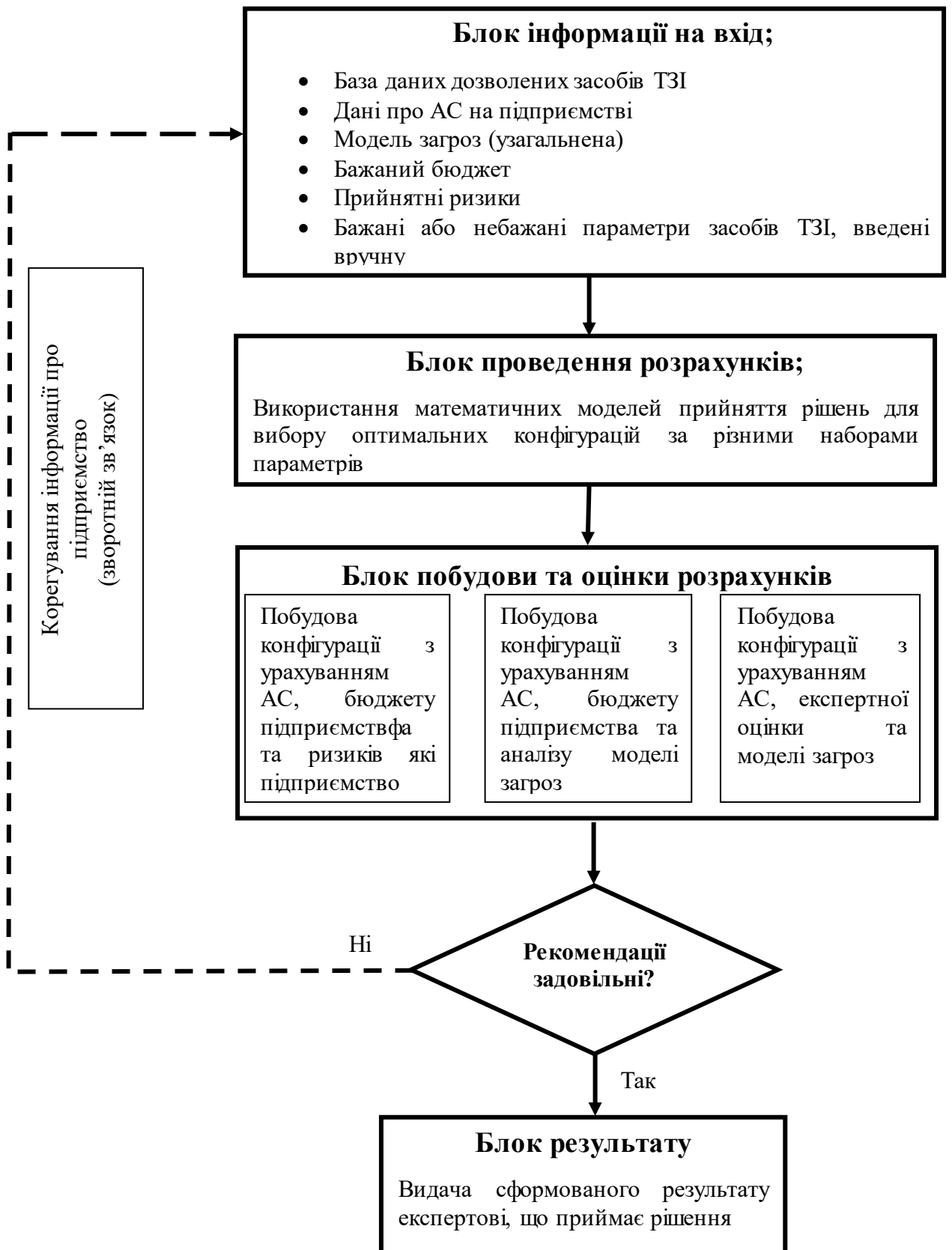


Рисунок 2.1 – Структурно-логічна схема роботи ІАС

У структурно-логічну схему увійшли чотири блоки, три блоки з чотирьох були зазначені вище в елементах структури:

- Блок інформації на вхід;
- Блок проведення розрахунків;
- Блок побудови та оцінки розрахунків;
- Блок результату.

Блок вхідної інформації включатиме дані, що заповнюватиме експерт про своє підприємство, а також БД дозволених ЗЗІ. Це ті дані, на основі яких ІАС формуватиме рекомендації, на основі яких прийматимуться усі подальші рішення. Блок розрахунків включатиме усі математичні методи, які застосовуватимуться для розрахунків оптимальних конфігурацій, та їхню практичну реалізацію. Блок формування оцінок та рекомендацій консолідуватиме в собі ТЗІ, що виявляться рекомендованими при одній з трьох конфігурацій. Блок видачі результату формуватиме читабельні звіти, куди входить вибраний експертом перелік рекомендованих ЗЗІ, на основі якого експерт будуватиме КСЗІ на підприємстві. У блоці проведення розрахунків проводяться розрахунки, які використовують математичну модель для прийняття рішення. Також береться до уваги конфігурація, обрана експертом, а також бажані параметри, які мають бути включені при розрахуванні. Блок побудови та оцінки результатів, має на меті викристалізувати рекомендації які допоможуть експерту в побудові КСЗІ. І останній блок результату формує звіт, який експерт буде використовувати під час прийняття рішення в побудові КСЗІ.

Також передбачений механізм зворотного зв'язку, що полягатиме у можливості змінити вхідні дані про підприємство, що потягне за собою зміни у виданих результатах.

Для розрахунків у розрахунковому блоці використовуватимуться математичні моделі прийняття оптимальних рішень, лінійне програмування та інші математичні розрахунки.

Подальша розробка ІАС вестиметься, виходячи з вищенаведеної структурно-логічної схеми.

2.3 Визначення елементів для розробки

Завдання цієї роботи передбачає побудову окремих елементів(компонентів) ІАС. Важливою задачею - є визначення тих елементів, що врешті-решт будуть розроблені в цій роботі.

З вищенаведеної структурної схеми, найдоцільнішими елементами для розробки є база даних дозволених засобів ТЗІ. Саме вона є результатом реструктуризації даних, поданих в офіційних переліках дозволених засобів ТЗІ, та прототипне програмне забезпечення, метою існування якого будуть: інтерфейс для взаємодії адміністратора з базою даних, тестування вводу конкретних побажань щодо засобів ТЗІ та виводу сформованого результату. Ці елементи, окрім започаткування розробки ІАС, зможуть повноцінно функціонувати всі разом та приносити практичну користь експертам, які зможуть власноруч на основі заданих фільтрів відбирати засоби ТЗІ та формувати кінцеві переліки ЗЗІ для створення КСЗІ.

Натомість, втілення блоку розрахунків не є доцільним у цій роботі, адже не може існувати поза ІАС. Як наслідок, втілення цього блоку як окремого елементу ІАС (включаючи визначення математичних моделей, на основі яких він формуватиметься) позбавлене сенсу і буде винесено на період розробки повної ІАС.

Блок формування оцінок та рекомендацій можливо втілювати лише після блоку розрахунків, адже його вивід повністю залежить від того, яким буде його ввід (тобто, результати, обраховані у блоці розрахунків). Як і з блоком розрахунків, втілення цього блоку буде винесено на період розробки повної ІАС.

Висновки до другого розділу

Після визначення необхідності побудови ІАС у першому розділі настала необхідність визначити конкретні напрями розробки ІАС.

Першим етапом стало визначення функціоналу, який забезпечуватиме ІАС. Було вирішено, що кінцеві рекомендації будуть надаватися не одним переліком, а трьома – для кожного з таких переліків використовуватиметься своя конфігурація вхідних даних. Завдяки цьому в експерта з'явиться простір для вибору між переліками, завдяки чому він зможе прийняти більш зважене рішення, ніж якби перелік рекомендованих ЗЗІ був безальтернативним.

Другим етапом стала побудова структурно-логічної схеми, в межах якої функціонуватиме ІАС. У ній чітко було позначені джерела вхідних даних, масиви вихідних даних та наявність можливості змінити дані, якщо результуючі рекомендації виявляться незадовільними для експерта.

Третім етапом стало визначення елементів ІАС, доцільних для втілення у даній роботі. Ними стали база даних як елемент вхідних даних ІАС та прототип ПЗ, на якому тестуватимуться ввід конкретних побажань щодо засобів ТЗІ від експерта (елемент вхідних даних та зворотнього зв'язку ІАС) та вивід кінцевого сформованого переліку (елемент вихідних даних ІАС).

Подальша робота полягатиме у практичному втіленні вищенаведених елементів ІАС.

РОЗДІЛ 3

РОЗРОБКА КОМПОНЕНТІВ ТА БАЗИ ДАНИХ ЯК ЕЛЕМЕНТУ ВХІДНИХ ДАНИХ ІНФОРМАЦІЙНО-АНАЛІТИЧНОЇ СИСТЕМИ

База даних є найважливішим елементом вхідних даних ІАС. Саме з переліку засобів ЗЗІ, що знаходяться в ній, відбиратимуться та обиратимуться засоби ТЗІ, необхідні для побудови КСЗІ.

У цьому розділі необхідно побудувати таку базу даних, яка передбачає конкретнішу класифікацію ЗЗІ, ніж в офіційних документах, та доповнити інформацію тією, що потенційно знадобиться експерту, що приймає рішення про відбір засобів ТЗІ.

3.1 Розробка бази даних. Інфологічна модель

Дані про ТЗІ, подані у відповідних переліках на офіційному сайті Державної служби спеціального зв'язку та захисту інформації України, містять [12, 13] наступну інформацію (порядок не збережено):

1. Назва засобу.
2. Тип засобу (наприклад, генератор шуму або антивірусне програмне забезпечення).
3. Властивість інформації, яку захищає пристрій: конфіденційність, цілісність чи доступність.
4. Програмний, апаратний чи програмно-апаратний комплекс.
5. Вимоги, яким відповідає засіб.
6. Компанія-виробник.

7. Сертифікат відповідності чи експертний висновок.
8. Функціональний профіль.
9. Рівень гарантій.
10. Категорія засобів КЗІ.
11. Додатком/компонентом чого є об'єкт.
12. Докладніший опис, для чого потрібен пристрій.

Дані, подані у таблицях, хоч і детальні, проте не вичерпні та слабо структуровані. У базі даних для одночасного спрощення подання інформації та її деталізації буде утворено декілька таблиць, що будуть містити додаткову інформацію про наступні аспекти: тип засобу; компанію-виробника; сертифікат відповідності чи експертний висновок; функціональний профіль; рівень гарантій.

База даних являє собою добірку інформації, систематизовану таким чином, щоб до неї можна було легко отримати доступ, управляти нею і оновлювати її. Комп'ютерні бази даних зазвичай містять агреговані дані або файли, що містять інформацію про операції купівлі-продажу або про взаємодію з конкретними клієнтами.

У реляційній базі даних цифрова інформація про конкретний клієнта організована в рядки, стовпці і таблиці, які індексуються, щоб полегшити пошук відповідної інформації через SQL або NoSQL запити. На відміну від цього, база даних графів використовує вузли і ребра для визначення зв'язків між входами даних і запитамі вимагає спеціального семантичного синтаксису пошуку. На момент написання SPARQL є єдиною мовою семантичних запитів, схваленим Консорціумом Всесвітньої павутини (W3C).

Як правило, менеджер бази даних надає користувачам можливість контролювати доступ до читання / запису, вказувати генерацію звітів і аналізувати використання. Деякі бази даних пропонують відповідність ACID

(атомічність, несуперечливість, ізольованість і довговічність), щоб гарантувати несуперечність даних і повноту транзакцій.

Бази даних розвивалися з моменту їх створення в 1960-х роках, починаючи з ієрархічних і мережових баз даних, з 1980-х років з об'єктно-орієнтованими базами даних, а сьогодні з базами даних SQL і NoSQL і хмарними базами даних.

З одного боку, бази даних можна класифікувати за типом змісту: бібліографічні, повні тексти, числа і зображення. При обчисленні бази даних іноді класифікуються відповідно до їх організаційним підходом. Існує безліч різних видів баз даних, починаючи від найбільш поширеного підходу, реляційної бази даних, і закінчуючи розподіленою базою даних, хмарної базою даних, графовою базою даних або базою даних NoSQL.

Реляційна база даних, винайдена Е.Ф. Коддом в IBM в 1970 році, являє собою табличну базу даних, в якій дані визначаються таким чином, що вони можуть бути реорганізовані і доступні різними способами.

Реляційні бази даних складаються з набору таблиць з даними, які відповідають зумовленою категорії. Кожна таблиця має щонайменше одну категорію даних в стовпці, і кожен рядок має певний екземпляр даних для категорій, які визначені в шпальтах.

Структурована мова запитів (SQL) - стандартний інтерфейс програми користувача і додатки для реляційної бази даних. Реляційні бази даних легко розширити, і нова категорія даних може бути додана після створення вихідної бази даних, не вимагаючи зміни всіх існуючих додатків.

Розподілена база даних - це база даних, в якій частині бази даних зберігаються в кількох фізичних точках і в якій обробка розосереджена або репліцирована між різними точками в мережі.

Розподілені бази даних можуть бути однорідними або гетерогенними. Всі фізичні локації в однорідній розподіленій системі баз даних мають однакове апаратне забезпечення і запускають одні і ті ж операційні системи і

додатки баз даних. Апаратні засоби, операційні системи або програми баз даних в гетерогенній розподіленій базі даних можуть бути різними в кожному місці.

Хмарна база даних - це база даних, яка оптимізована або побудована для виртуалізованого середовища, або в гібридному хмарі, громадському хмарі або приватному хмарі. Хмарні бази даних забезпечують такі переваги, як здатність платити за ємність пам'яті і пропускну здатність в залежності від виду використання, а також можливість розширення на вимогу поряд з високою доступністю.

Хмарна база даних також дає підприємствам можливість підтримувати бізнес-додатки при розгортанні програмного забезпечення в якості послуги.

Бази даних NoSQL корисні для великих наборів розподілених даних.

Бази даних NoSQL ефективні в питаннях продуктивності великих даних, які реляційні бази даних не призначені для вирішення. Вони найбільш ефективні, коли організація повинна проаналізувати великі шматки неструктурованих даних або даних, які зберігаються на декількох віртуальних серверах в хмарі.

Елементи, створені з використанням об'єктно-орієнтованих мов програмування, часто зберігаються в реляційних базах даних, але об'єктно-орієнтовані бази даних добре підходять для цих елементів.

Об'єктно-орієнтована база даних організована навколо об'єктів, а не дій, а дані, а не логіка. Наприклад, мультимедійна запис в реляційній базі даних може бути певним об'єктом даних, на відміну від буквено-цифрового значення.

База даних, орієнтована на графи, або база даних графів, є типом бази даних NoSQL, яка використовує теорію графів для зберігання, відображення і запитів. Бази даних графів по суті є колекціями вузлів і ребер, де кожен вузол являє сутність, а кожне ребро представляє з'єднання між вузлами.

Бази даних графів стають все більш популярними для аналізу взаємозв'язків. Наприклад, компанії можуть використовувати графічну базу даних для пошуку даних про клієнтів в соціальних мережах.

Бази даних графів часто використовують SPARQL, декларативний мову програмування і протокол для аналізу баз даних графів. SPARQL має можливість виконувати всі аналітичні функції, які може виконувати SQL, плюс він може бути використаний для семантичного аналізу, вивчення взаємозв'язків. Це дозволяє проводити аналіз наборів даних, що мають як структуровані, так і неструктуровані дані. SPARQL дозволяє користувачам виконувати аналіз інформації, що зберігається в реляційній базі даних, а також відносин одне одного (FOAF), PageRank і найкоротшого шляху.

Основними конструктивними елементами інфологічної моделі є сутність, зв'язки між ними та їх властивості (атрибути).

Сутність - це весь ідентифікований об'єкт, який повинен зберігатися в базі даних (його можна відрізнити від іншого). Люди, місця, літаки, польоти, задоволення, кольору і т.д ... Ви повинні відрізнити тип сутності від екземпляра сутності. Тип таблиці макіяжу, однорідних людей, об'єктів або подій відноситься до цілого контролера. Приклад об'єкта відноситься до чогось специфічного в наборі. Наприклад, місто правосуб'єктності, наприклад, Полтава, Київ і т.д. може їм бути. Він може бути унікальним для певного типу правосуб'єктності, але може бути однаковим для різних типів об'єктів (наприклад, він може бути кольором запаху багатьох речей, таких як собаки, машини, дим).

Атрибути використовуються для визначення того, яку інформацію слід збирати про солодощі. Червоний, синій, банан і т.д. Існує багато станів або цінностей знайти потрібний екземпляр об'єкта. Мінімум означає, що він не дозволяє визначити ідентифікатор, ґрунтуючись на тому факті, що чіп будь-якого набору атрибутів є ланцюговим.

Зв'язок є асоціацією двох або більше сутностей. Якщо ми зберігаємо тільки окремі і незв'язані дані, структура даних буде дуже простий. Однак

одним з основних вимог для організації реляційної бази даних є можливість перегляду одного об'єкта на основі цінностей інших, і між ними повинні бути встановлені певні зв'язки. Оскільки фактичні бази даних теоретично мають сотні об'єктів, ці активи можуть бути пов'язані між собою. Наявність такої безлічі зв'язків і визначає складність інфологічної моделі.

Особливо корисним буде інформація про функціональні профілі та рівні гарантій, бо вона розкриває корисний функціонал засобів інформації та чіткіше дозволяє визначити необхідність їхнього використання.

Деякі з додаткових даних через їхню сутність (як-то репутація компанії-виробника) не відображені в офіційному документі та заповнюватимуться експертом, що підтримуватиме актуальність бази даних. Також до таблиці буде додана ринкова вартість засобу, яка теж не має відображення в офіційних переліках через характер цих документів. Враховуючи все вищенаведене, була складена інфологічна модель бази даних, яка представлена на рис. 3.1.

Варто зазначити, що пункт 11 заповнюється лише у тому випадку, коли пристрій не є самостійним та доповнює функціонал іншого пристрою. [17]

При побудові інфологічних моделей можна використовувати мову *ER-діаграм* (від англ. Entity-Relationship, тобто сутність-зв'язок). У них є певні позначення:

Зв'язки позначаються стрілочками, над якими має проставлятися ступінь зв'язку (1 або буква, яка замінює слово “багато”) і додаткове пояснення.

Між двома сутностями, наприклад, А і В, чотири види зв'язку можливі.

1. Зв'язок один-до-одного (1:1): у момент періодичного часу кожному екземпляру сутності А відповідає 1 або 0 екземпляр сутності В.

2. Зв'язок один-до-багатьох (1:М): одному представнику сутності А відповідають 0, 1 або декілька представників сутності В.

3.2 Розробка бази даних. Даталогічна модель

Кожній таблиці інфологічної моделі відповідає таблиця даталогічної моделі. Вигляд таблиць, утім, зазнав змін, виходячи з реальних потреб розміщення даних у БД.

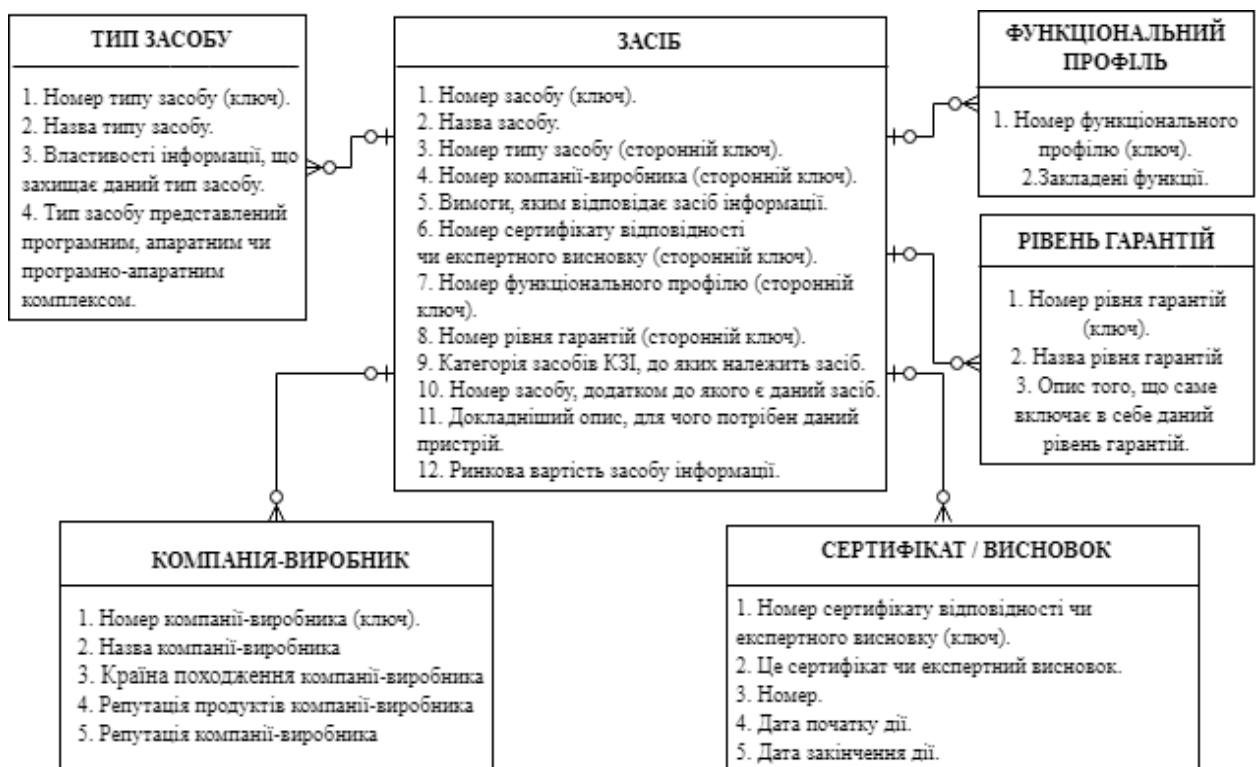


Рисунок 3.1 – Інфологічна модель

У випадку таблиці «Засіб» замість номера функціонального профілю буде використовуватися рядок із переліченими функціями. Так було перероблено для спрощення структури бази даних та через те, що виробники часто ігнорують стандартні функціональні профілі, визначені в НД ТЗІ 2.5-005-99, та створюють власні, просто перелічуючи наявний у продукті функціонал.

Модель системи реєстрації даних являє собою модель логічного рівня і відображає логічні взаємозв'язки між елементами даних незалежно від їх змісту і середовища зберігання. Ця модель була створена для інформаційних одиниць, дозволених в СУБД, де ми розробили базу даних. Етап створення моделі системи реєстрації даних називається проектуванням системи реєстрації даних. Опис логічної структури бази даних на мові СУБД називається діаграмою.

Хоча конструкція системи реєстрації даних є логічною структурою бази даних, на неї впливає фізична організація даних, що надаються конкретною СУБД. Тому інформація про організацію фізичних даних є корисною при розробці логічної структури. Логічна структура бази даних і бази даних відображає фактичну тематичну область. Таким чином, специфіка тематичної області, показана в комп'ютерній моделі, безпосередньо впливає на вибір дизайнерських рішень.

Для реляційної бази даних логічного проектування структура складається з синтаксичного аналізу всієї файлової інформації (взаємозв'язків), а також визначення складу області (атрибутів) для кожного з цих файлів. Однією з вимог є стандартизація реляційної бази даних. Процес стандартизації спрямований на усунення надмірності даних і стандартизацію третього нормального формату (або звичайної форми Size-Codd). Нормалізація являє собою офіційну процедуру (БД), в рамках якої елементи групуються в таблиці і таблиці.

Аналіз тематичної області, як правило, проводиться на основі даних, відомих в цьому контексті, з урахуванням цілей розробки програмної системи. Бази даних називаються електронними сховищами даних, до яких можна отримати доступ за допомогою одного або декількох комп'ютерів. Зрозуміло, база даних створюється для зберігання і отримання доступу до

даних, що містить інформацію про конкретну тематичної області, тобто області діяльності людини або області реального світу.

Для створення реляційних баз даних логічної структури: Вся інформація про файлах (взаєминах) і складі полів (атрибутів) для кожного з цих файлів включена. Одиницею інформації, що зберігається в базі даних, є таблиця. Кожна таблиця є набором рядків і стовпців (властивості, властивості, параметри об'єкта, подія, явище), де рядки відповідають екземпляру, а стовпці відповідають атрибутам.

У випадку таблиці «Компанія-виробник» поля «репутація продуктів» та «репутація компанії» продубльовані двічі – для числового значення від 1 до 10 та для текстових коментарів-пояснень до цих цифр. Також додано поле «Повна назва компанії та її адреса» для зазначення повної назви юридичної особи компанії та її адреси (зазвичай у переліках подається юридична особа української філії виробника).

На етапі розробки даних відбувається перехід від програмної комп'ютерної моделі до логічної моделі (реєстрація даних), яка підтримується деякими СУБД. Перехід від інфрачервоної моделі до системної моделі реєстрації даних називається побудовою. Модель системи реєстрації даних являє собою базу даних, яка логічно конфігурується і фокусується на конкретній СУБД. Для створення системи реєстрації даних необхідно вибрати СУБД. Кожна окрема СУБД встановлює ряд обмежень на структуру логічної моделі даних, тому в першу чергу потрібно

Огляд специфічності і специфічності СУБД, виявлення всіх факторів, які можуть впливати на логічну модель БД.

Основними факторами, що впливають на структуру системи реєстрації даних в СУБД, є:

1. Вибирається тип логічної моделі, підтримуваний СУБД. СУБД є реляційним, найбільш поширеним програмним забезпеченням і

автоматизацією економічних розрахунків на ринку. На додаток до реляційних моделей, бази даних мають ієрархічні і мережні моделі.

2. Фізична організація даних в вибраному середовищі організована в базі даних СУБД.Napriclade Paradox або базі даних системи dBASE у вигляді DT і DBF пов'язаних форматів файлів, як і всі інші об'єкти, форми та звіти. Дані та інструменти зберігаються в одному файлі бази даних.

Тому при розробці бази даних необхідно знати не тільки правила логічної структури, а й характеристики фізичної організації бази даних. Кількісні обмеження, що застосовуються (наприклад, число ієрархічних рівнів, кількість полів, кількість записів, кількість файлів в ієрархічних моделях).

До сих пір не знайдений формальний метод, що дозволяє розробляти унікальний дизайн системи реєстрації даних. Таким чином, результат в значній мірі залежить від кваліфікації та кваліфікації експертів, які будуть займатися розробкою проекту. В результаті проектування системи реєстрації даних можуть бути отримані різні варіанти побудови логічної моделі даних. Тому важливо оцінити отримані моделі і вибрати найкращий варіант. Першим кроком є оцінка результатів на предмет наявності машини.

Якщо це обмеження не дотримується, БД може бути змінена. Крім того, повинні бути перевірені вимоги щодо дотримання всіх вимог, що пред'являються користувачем, і вимог, що пред'являються до додатків в імпортованій моделі, тобто логічної моделі тематичної інформаційної моделі. Вимоги які висуваються в процесі побудови БД повинні бути виконані. В результаті проектування системи реєстрації даних можуть бути отримані різні варіанти побудови логічної моделі даних. На додаток до реляційних моделей, бази даних мають ієрархічні і мережні моделі.

Таблиця «Функціональний профіль» була замінена таблицею «Функції», де зберігаються аббревіатури та пояснення функцій засобів захисту,

відповідно до НД ТЗІ 2.5-005-99, через надмірність інформації, що зберігалася б у таблиці «Функціональний профіль». Новостворена таблиця не матиме зв'язку з таблицею «Засіб» (причини вказано під табл. 3.4).

Кожному полю присвоюється тип даних, який буде використовуватися для зберігання інформації, що міститиметься в полі. Таблиці виглядають наступним чином:

Таблиця 3.1

Таблиця «Засіб»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер засобу (ключ)	INTEGER	ні
Назва засобу	TEXT	ні
Номер типу засобу (сторонній ключ)	TEXT	ні
Номер компанії-виробника (сторонній ключ)	INTEGER	ні
Вимоги, яким відповідає засіб захисту	TEXT	ні
Номер сертифікату відповідності чи експертного висновку (сторонній ключ)	INTEGER	ні
Функціональний профіль	TEXT	так
Номер рівня гарантій (сторонній ключ)	INTEGER	так
Офіційна категорія засобів КЗІ, до яких належить засіб	TEXT	ні
Номер засобу захисту (або іншого продукту), додатком до якого є даний засіб	INTEGER	так
Докладніший опис, для чого потрібен даний пристрій	TEXT	ні
Ринкова вартість засобу захисту	INTEGER	так

Як правило, дані мають бути заповнені обов'язково, тому відсутність значення у комірці зазвичай заборонена. Втім, є деякі виключення, які варто навести з аргументацією.

У цій таблиці можуть бути не вказані:

- функціональний профіль – через те, що не всі виробники зазначають функціональний профіль власних виробів;
- рівень гарантій – через те, що не всім виробам присвоюється певний рівень гарантій;
- номер засобу захисту, додатком до якого є даний засіб захисту – через те, що засіб захисту може бути (і зазвичай є) самостійним.

Таблиця 3.2

Таблиця «Тип засобу»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер типу засобу (ключ)	TEXT	ні
Назва типу засобу	STRING	ні
Властивості інформації, що захищає даний тип засобу	STRING	ні
Тип засобу представлений апаратним, програмним чи апаратно-програмним комплексом	INTEGER	ні

Властивості інформації утворюють обмежену кількість комбінацій, тому з точки зору використання пам'яті легше дати цим комбінаціям відповідні числові значення, ніж у кожному окремому випадку задавати текстове значення властивостей. Аналогічно з тим, як представлений засіб захисту певного типу. Адже всі ці типи мають у собі певний перелік інформацій, які не можуть бути використані перехресно. Під час виконання розрахунків, буде використовуватись обмежена кількість дозволених символів, за для того щоб зберегти пам'ять та легше комбінувати числові значення. Як правило, дані мають бути заповнені обов'язково, тому відсутність значення у комірці зазвичай заборонена.

Таблиця 3.3

Таблиця «Компанія-виробник»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер компанії-виробника (ключ)	INTEGER	ні
Назва компанії-виробника	TEXT	ні
Повна назва компанії-виробника та її адреса	TEXT	ні
Країна походження компанії-виробника	TEXT	ні
Репутація продуктів виробника (текстове пояснення)	TEXT	так
Репутація компанії-виробника (текстове пояснення)	TEXT	так
Репутація продуктів виробника (число)	INTEGER	ні
Репутація компанії-виробника (число)	INTEGER	ні

Дозволяється не вносити текстових пояснень.

До останніх двох числових полів можна вводити лише значення від 1 до 10.

Таблиця 3.4

Таблиця «Функції»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер функції (ключ)	INTEGER	ні
Абревіатура	TEXT	ні
Розшифровка абревіатури	TEXT	ні

У цій таблиці представлені абревіатури функціоналу та їхні розшифровки, згідно з НД ТЗІ 2.5-005-99.

У цій таблиці немає зв'язку з основною таблицею «Засіб» через те, що у тій таблиці вказується повний функціональний профіль, натомість у цій таблиці містяться лише поодинокі функції, з множини яких і складається функціональний профіль.

Таблиця 3.5

Таблиця «Сертифікат / Висновок»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер (у БД) сертифікату чи експертного висновку (ключ)	INTEGER	ні
Сертифікат чи експертний висновок	BOOLEAN	ні
Номер документу	STRING	ні
Дата початку дії	BLOOB	ні
Дата закінчення дії	BLOOB	ні

Для поля «номер документу» обраний рядковий тип, враховуючи, що в це поле фактично буде занесена повна назва документа (головною частиною якої є номер) та що математичні дії до номерів сертифікатів застосовуватися не будуть.

Таблиця 3.6

Таблиця «Рівень гарантій»

Назва рядку	Тип даних	Дозволена відсутність значення
Номер рівня гарантій (ключ)	INTEGER	ні
Абревіатура	TEXT	ні
Пояснення	TEXT	ні

Поля цієї таблиці не потребують пояснення.

Окрім таблиці «Функції», інші відношення таблиць дублюють відношення в інфологічній моделі даних.

Після складення даталогічної моделі даних треба реалізувати її на сервері реляційної бази даних.

3.3 Розробка бази даних. Втілення моделі на сервер реляційної бази даних

На початку, звертаючи увагу на кінцеву мету, що передбачає структурування даних, варто відмовитися від нереляційних баз даних на користь NoSQL на користь реляційних. Чому вибір припав саме на реляційну базу даних, а не на нереляційну, спробую пояснити нижче.

SQL також відомий як реляційні бази даних і як бази даних, які не пов'язані з базами даних NoSQL. Неперіодичні бази даних зазвичай орієнтовані і поширюються в вигляді документів, а не структуровані. Сьогодні використовується безліч різних типів баз даних. Данські бази даних SQL були і залишаються найбільш поширеними базами даних. Деякі з них популярні в MySQL, Oracle та деяких інших типах баз даних SQL.

Однак, бази даних NoSQL, такі як MongoDB, недавно стали популярними. Бази даних SQL і NoSQL використовують ту ж частину поштової скриньки, яка повинна вести себе як база даних; наприклад, вони збирають і впорядковують атрибути. SQL баз даних не були розроблені в 1974 році, коли вони існували в 1960 році

За рік. Всі бази даних преоплативності SQL прийшли в NoSQL Руднайн. SQL також відомий як реляційні бази даних і є більш структурованим, ніж бази даних NoSQL, відомі як неасоціювані бази даних. Неперіодичні бази даних часто більшою мірою орієнтовані на документи і поширюються в порівнянні з створеними базами даних.

Наприклад, ви можете сказати, що бази даних, що не стосуються справи, такі як телефонні книги, в яких зберігаються телефонні номери і адреси

реляційних баз даних, схожі на папки з файлами, які містять всі, від адрес і телефонних ліній до Facebook і онлайн -покупок. Основна відмінність між ними полягає в використанні SQL, мови запитів, який конфігурується як стандартизований мову програмування для керування базами даних баз даних SQL і виконання різних операцій з даними в цих базах. Бази даних NoSQL, як правило, не використовують табличні відносини для зберігання і пошуку даних, які є загальними для реляційних баз даних. Замість цього значення - це документи, які зберігають пари ключів, графічні бази даних або зберігають з великими стовпцями.

NoSQL в першу чергу зосереджений на зборі документів. Крім того, NoSQL ідеально використовується для зберігання великого набору даних, зазвичай зберігаються в документах JSO. В основному, коли бази даних SQL представляють дані в форматі таблиць, бази даних NoSQL відображають пару ключових значень, документи, графічні бази даних або сховища великих стовпців, всі з яких не мають стандартних визначень схеми. З іншого боку, ви повинні слідувати заздалегідь визначеною схемою баз даних SQL.

Таблиця 3.7

Порівняння SQL та NoSQL БД

	База даних NoSQL	База даних SQL
Також відомий як	Нереляційна або розподілена база даних	Реляційні бази даних (RDBMS)
Мова	Запити зосереджені на зборі документів. Іноді його також називають UnQL (Unstructured Query Language).	Використовуйте SQL (Structured Query Language) для визначення та маніпулювання даними
Основа	Бази даних є документами, парами ключ-значення,	Бази даних на основі таблиць

	базами даних графів або магазинами з широкою колоною	
--	--	--

продовження табл. 3.7

Зберігання даних	Ієрархічне зберігання даних	Немає ієрархічного зберігання даних
Схема	Майте динамічну схему для неструктурованих даних	Маю заздалегідь визначену схему
Схема	База даних є схема-агностик і диктується програмою. Вона дозволяє здійснювати спритність і високу ітеративність розвитку	Схему потрібно підтримувати і зберігати в синхронізації між програмою та базою даних
Властивості даних	Пропонує гнучкість, оскільки не всі записи повинні зберігати ті ж самі властивості	Відмінно підходить для рішень, де кожен запис має однакові властивості
Нові дані	Нові дані можуть бути додані без будь-яких застережень	Додавання нових даних може потребувати зміни схем або засипання даних
Типи даних	Добре для напівструктурованих, складних або вкладених даних	Добре для структурованих даних
Відносини	Відносини між даними часто захоплюються шляхом денормалізації даних і представляють всі дані для об'єкта в одному записі	Відносини часто фіксуються в нормалізованих моделях, використовуючи об'єднання для вирішення посилань між таблицями
Масштабованість	Розширюються горизонтально.	Вертикально масштабовані. Масштабуються за рахунок

	Масштабуються шляхом збільшення серверів баз даних у пулі ресурсів для зменшення навантаження.	збільшення потужності апаратного забезпечення.
--	--	--

продовження табл. 3.7

Комплексні запити	Не підходять для складних запитів	Добре підходять для складного запиту інтенсивного середовища
Стандартні інтерфейси	Не використовуйте стандартні інтерфейси для виконання складних запитів	Майте стандартні інтерфейси для виконання складних запитів
Розмір набору даних	Дуже кращий для великого набору даних, як може зберігати ієрархічні дані	Не ідеальний для великого набору даних
Комплексні транзакційні додатки	Не порівнянні і досить стабільні при високому навантаженні і для складних транзакційних додатків.	Краще підходить для важких транзакційних типів додатків, оскільки він більш стабільний і обіцяє атомність, а також цілісність даних.
Прийняття та підтримка	Не набули широкого розповсюдження і лише підтримка місцевих громад	Широко прийнята та легша підтримка
Властивості	Дотримується теорема CAP з пивоварів (послідовність, доступність і толерантність до розділів). Деякі бази даних можуть слідувати властивостям ACID.	Підкреслює властивості ACID (атомність, консистенція, ізоляція та довговічність)
Приклади	MongoDB, BigTable, Redis, RavenDb, Cassandra, Hbase, Neo4j і CouchDb	MySQL, Oracle, Sqlite, Postgres і MS-SQL

На додаток до цих, існують різні інші відмінності між ними. Перевагою NoSQL над SQL є той факт, що він є більш змінним і може бути адаптований до будь-якої потреби. В той час як SQL, як правило, був більш структурованим і менш адаптованим.

Потім варто вибрати реалізацію реляційної бази даних. Основними найпопулярнішими реалізаціями на 2021 рік [18] є, у порядку спадання популярності, Oracle, MySQL, Microsoft SQL Server, PostgreSQL та MongoDB. Переваги, що надаються одним втіленням перед іншим, не є суттєвими для цієї роботи. Як і у випадку з вибором мови програмування, цей вибір не є суттєвим, тому був обраний SQLite завдяки власному досвіду роботи з ним.

SQLite - це проста реляційна система управління базами даних. Це бібліотека, в якій використовується більшість стандартів SQL-92. Вихідний код SQLite поширюється як публічна область (Public domain), що означає, що він може використовуватися безкоштовно в будь-яких цілях і без будь-яких обмежень. Фінансова підтримка розробників SQLite, Adobe, Oracle, Mozilla, Nokia, забезпечується спеціально створеним консорціумом компаній, таких як Bentley і Blomberg. З 2018 року SQLite як JSO і SV рекомендується в якості формату для зберігання набору даних, налаштованих Бібліотекою Конгресу.

Проект отримав нагороду в 2005 році. Функція SQLite не використовує парадигму клієнт-сервер, тобто движок SQLite не є окремим процесом для додатка для взаємодії, але показує, яка бібліотека програми в програмі. Тому, як протокол змін використовує виклики функцій (API) з бібліотеки SQLite. Такий підхід скорочує додаткові витрати і час реагування і спрощує програму. SQLite зберігає всю базу даних (включаючи визначення, таблиці, каталоги і дані) в єдиний стандартний файл на машині, на якій виконується додаток.

Простота додатки полягає в тому, що всі файли, що зберігають базу даних, блокуються перед виконанням операції шляхом створення логарифмічного файлу, а також функцій ACID. Кілька процесів або потоків можуть легко читати дані з однієї і тієї ж бази даних одночасно. Ви можете писати в базу даних тільки тоді, коли інші запити не зберігатимуться, інакше спроба запису провалиться і код помилки повернеться в програму. Інший варіант полягає в автоматичному повторенні спроб запису протягом даного періоду часу.

Пакет також включає функціональну частину клієнта в квадратний виконуваний формат 3, який показує реалізацію основних бібліотечних функцій. Клієнтська частина працює на звичайній лінії і дозволяє посилатися на файл бази даних на основі типових функцій операційної системи.

Завдяки архітектурі двигуна, SQLite можна використовувати на вбудованих (вбудованих) і спеціальних машинах з гігабайтними наборами даних.

Особливості SQLite:

Операції навіть фіксуються в атомних, послідовних, ізольованих, надійних (AQID) системах - установка не потрібно, додаток управління є важливою частиною стандарту SQL92 [2, база даних зберігається в межпространственном файлі диска

Розміри баз даних Terabyte і Gigabyte Rosemoors і розмір коду BLOBуv Malius: Повністю сконфігуровані і менш ніж mensh200KB для клієнтських баз даних з додатковими функціями, ніж Schweed, і популярний сервер для простої налагодження API, записаний на ANSI c, підключений до TCL; доступний для десятків інших Коментарів MoBre Вихідний код% 100 прапорців можна використовувати як один вихідний файл в ANSI C, легко додаючи його в інший Autonomist проект: Крос-платформа без зовнішньої підписки: з коробки, підтримуваної Windos. Системи Ладанов Sirtsі легко

завантажуються в глобальне підмножина з окремим клієнтом інтерфейсу командного рядка, який може бути використаний для управління базами даних:

- транзакції атомарні, послідовні, ізольовані, і міцні (ACID) навіть після збоїв системи і збоїв живлення
- Встановлення без конфігурації — не потребує ані установки, ані адміністрування
- База даних зберігається в одному крос-платформовому файлі на диску
- Підтримка терабайтних розмірів баз даних і гігабайтного розміру рядків і BLOBів
- Швидший за популярні рушії клієнт-серверних баз даних для найпоширеніших операцій
- Доступний як єдиний файл сирцевого коду на C, який можна легко вставити в інший проєкт
- Автономність: немає зовнішніх залежностей
- Сирці перебувають в суспільному надбанні

Таблиця 3.7

Відповідні абстрактним типам даних конкретні типи даних SQLite

Абстрактний тип даних	Тип даних SQLite
NULL	Значення - NULL.
INTEGER	це підписане ціле число, яке зберігається в 1, 2, 3, 4, 6 або 8 байтах залежно від величини значення.
REAL	це значення з плаваючою точкою, яке зберігається як 8-байтове число з плаваючою точкою IEEE.
TEXT	це текстовий рядок, що зберігається з використанням кодування бази даних (UTF-8, UTF-16BE або UTF-16LE).

BLOB

що зберігається точно так, як була введена.

Клас зберігання є більш загальним, ніж тип даних. Наприклад, клас зберігання INTEGER включає 6 різних цілих типів даних різної довжини. Це робить різницю на диску. Але як тільки значення INTEGER зчитуються з диска та в пам'ять для обробки, вони перетворюються на найзагальніший тип даних (8-байтове ціле число з підписом). І тому здебільшого "клас зберігання" неможливо відрізнити від "типу даних", і ці два терміни можна використовувати як взаємозамінні.

Будь-який стовпець бази даних SQLite версії 3, крім стовпця INTEGER PRIMARY KEY, може використовуватися для зберігання значення будь-якого класу зберігання. Усі значення в операторах SQL, будь то літерали, вбудовані в текст оператора SQL, або параметри, прив'язані до попередньо скомпільованих операторів SQL, мають неявний клас зберігання.

Діаграма стосунків (рис. 3.2) була побудована наступним чином:

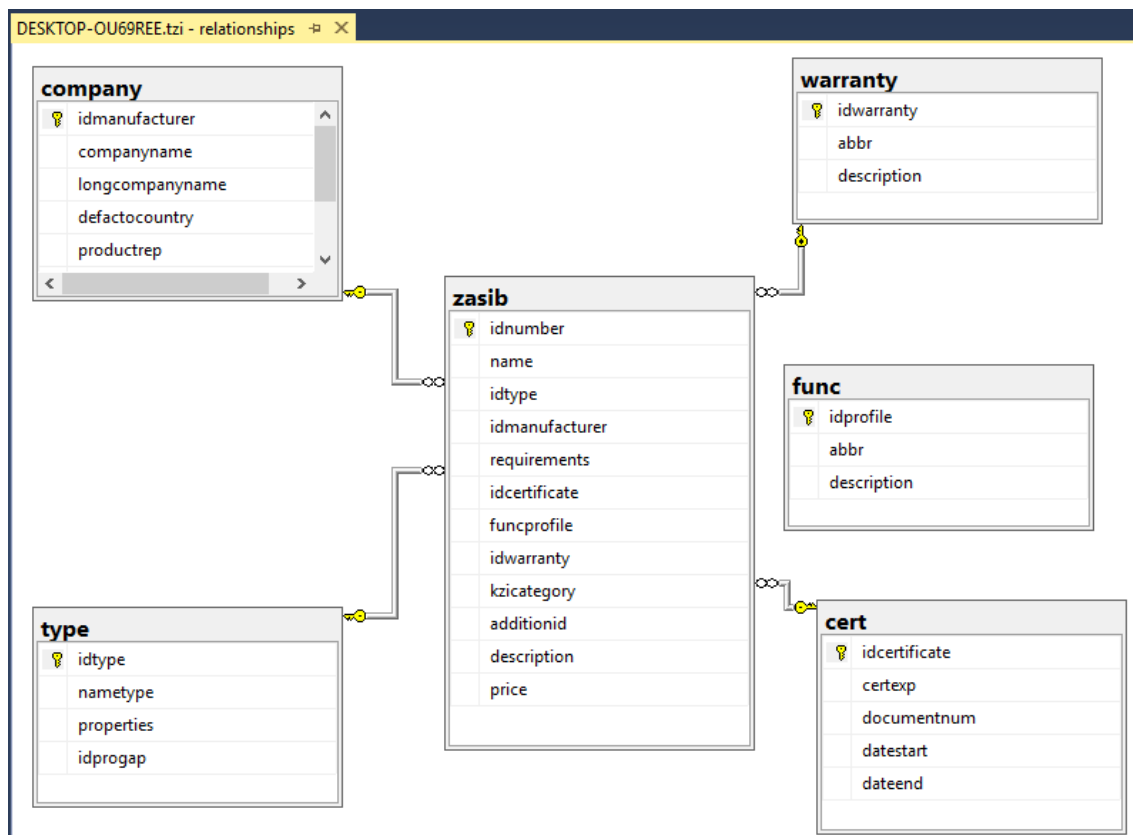


Рисунок 3.2 – Втілена на БД діаграма стосунків

На цьому завершився етап втілення даталогічної моделі БД. SQL-код для створення такої бази даних наведено у Додатку А. Після цього етапу БД треба наповнити змістом.

Зміст цілком взятий з офіційних документів ДССЗЗІ, адже їх опрацювання є присутнє в меті цієї роботи. Перелік документів, що висвітлюють дозволені ЗЗІ, були висвітлені у підрозділі 1.3 цієї роботи.

Певні дані з таблиці не були додані до бази даних через їх надлишковість. Вони жодним чином не вплинуть на кінцеве рішення експерта. До таких даних належать, наприклад, номери завдань (проектів), видані вітчизняним виробам.

Наступною частиною програми буде використання та створення програми контролю БД.

Висновки до третього розділу

Була створена база даних ЗЗІ, дозволених українським законодавством для використання у КСЗІ, шляхом втілення на сервер реляційної бази даних SQLite даталогічної моделі та наповнення її даними з офіційних документів ДССЗЗІ.

База даних придатна для використання майбутньою інформаційно-аналітичною системою вибору та містить повний обсяг інформації, необхідний експертові для прийняття рішення про створення КСЗІ. Враховуючи більш жорстку категоризацію інформації у базі даних порівняно з таблицею, із таблиці були вилучені незначні надлишкові дані, які не заважатимуть прийняттю рішень експертом.

Ця база даних слугуватиме джерелом вхідних даних для прототипного програмного забезпечення та майбутньої інформаційно-аналітичної системи.

РОЗДІЛ 4

ДОСЛІДЖЕННЯ РОБОТИ ПРОТОТИПУ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ВВОДУ ДАНИХ ТА ПРЕДСТАВЛЕННЯ РЕЗУЛЬТАТІВ

Після створення бази даних, потрібно створити й ПЗ, що з нею взаємодіє.

Прототип ПЗ, як було визначено в розділі 2, міститиме наступний функціонал:

- інтерфейс адміністратора для взаємодії з базою даних;
- ввід експертом побажань щодо засобів ТЗІ (і фільтрація переліку на основі цих побажань) як елемент блоку вхідної інформації ІАС;
- вивід переліку обраних засобів ТЗІ як елемент блоку видачі результатів ІАС.

4.1 Обґрунтування вибору мови і системи програмування

На сьогоднішній день існує безліч операційних систем які використовують комп'ютери. Робити системи для певних операційної системи, призведе до того, що певна кількість користувачів не матиме доступ до програми. Робити програму під всі платформи потребує багато зусиль та часу. Саме тому, було прийнято рішення створити веб-додаток. Для використання програми знадобиться веб-браузер, який за замовчуванням встановлений у більшості операційних систем. Доступ до ІАС зможе

отримати будь-який користувач який користується браузерами Google Chrome, Mozilla Firefox, Opera та інші. Це допоможе звільнити спеціалісти від використання певної операційної системи.

Під час підготовки до написання програми, була обрана мова програмування – Python. Ця мова протягом багатьох років проявила себе як надійна та швидко розвиваючися. Ця мова має безліч прихильників по всьому світу, які покращують її. Також існує безліч бібліотек які можуть допомогти виконати будь-які завдання.

Базуючись на мові програмування, потрібно обрати базу даних, яку буде наповнювати експерт та користувачи. База даних має бути проста в налаштуванні та повністю виконувати поставлені перед нею цілей. Провівши певні дослідження була обрана база даних – SQLite. Ця база даних має докладну документацію з повним поясненням функціоналу та можливостей. Також вона чудово інтегрується з мовою програмування – Python.

Після обрання бази даних потрібно обрати яким чином спеціаліст зможе реалізувати взаємодію з ІАС. В такому випадку розробка має відбуватись з використанням технологій, які використовують сучасні браузери. А саме HTML та CSS. Це є фундаментальні мови на яких будуються усі сайти в інтернеті. З плином часу були розроблені більш складні мови програмування які дозволяють покращувати роботу з сайтом та розширювати його функціонал. Таким чином використовують мову програмування JavaScript. Проте, для написання сайту з використанням цієї мови, потрібні навички та час на її вивчення. Тому, був направлений зір на фреймворки на базі Python. Був обраний фреймворк Django. Це потужний фреймворк, який використовує безліч компаній по всьому світі. На ньому побудовані наступні сайти: Dropbox, Spotify, YouTube, Instagram та інші. Саме тому вибір фреймворка є доцільним та не має жодних сумнівів що цілі, поставлені у цій роботі, будуть досягнуті завдяки ньому.

4.2 Обґрунтування вибору середовища програмування

Після того, як була обрана мова програмування, потрібно обрати середовище розробки програмного забезпечення. Найпопулярніша серед розробки, також її називають IDE(англ. integrated development environment - інтегроване середовище розробки) комплексне програмне рішення для побудови програмного забезпечення. Складається з редактора, інструментів для автоматизації та відлагодження програм. Також такі системи мають функцію автодоповнення коду. Існує середовище розробки PyCharm, створене для полегшення людям які програмають на мові Python.

Також був обраний графічний інтерфейс для контролю та аналізу бази даних SQLite – DB Browser. Проста в використанні програма по роботі з базами даних

4.3 Опис програми та її алгоритми

Програма складається з двох частин “Панель Адміністратора” та “Панель користувача.” Панель адміністратора була розроблена для використання адміністратором БД. Панель має всі інструменти для додавання, зміни та видалення записів з таблиць в базі даних.

Панель користувача обладнена можливістю використовувати засоби для сортування та отримання потрібних відомостей про певні ЗЗІ, які потрібні експерту за побудови КСЗІ.

4.3.1 Панель адміністратора

Панель становить точку керування базою даних шляхом її керування

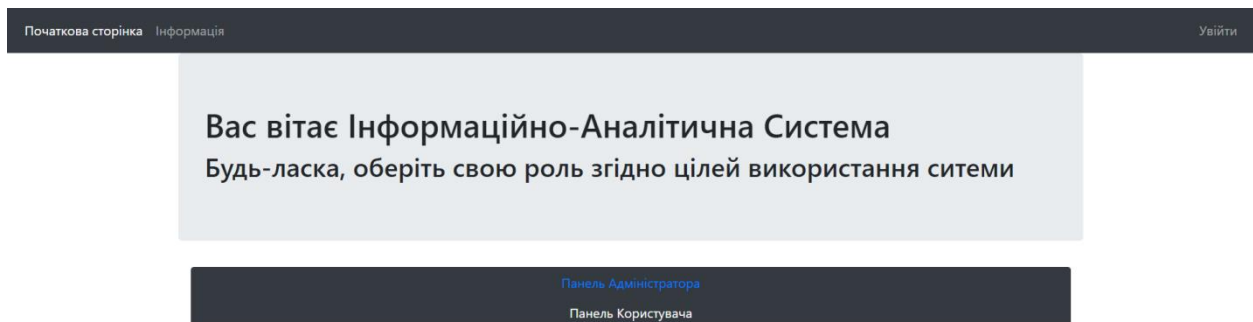


Рисунок 4.1 - Початкова сторінка інформаційно-аналітичної системи

Панель адміністратора виглядає як СУБД без можливості створення нових таблиць. Адміністратор має право їх редагувати, видаляти записи або додавати записи.

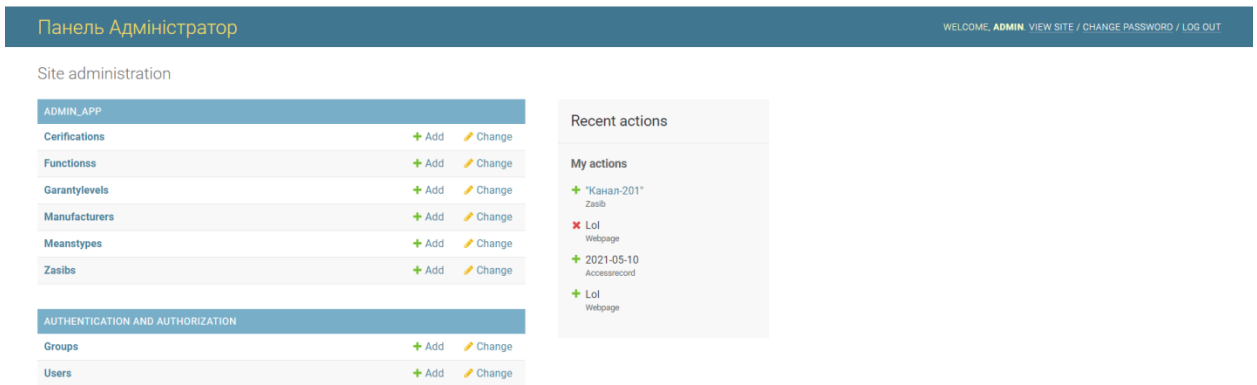


Рисунок 4.2 - Панель адміністратор

На рис 4.2 зображення панель адміністратор. Адміністратор має доступ в таблиці бази даних під назвою "ADMIN_APP". Також адміністратор може створювати групи та користувачів. Ці користувачі такоюж можуть мати повний доступ до даних, частковий або зовсім не мати. Це все регулюється шляхом організації груп (рис 4.3). У цій панелі адміністратор назначає права редагування, читання, додавання та видалення.

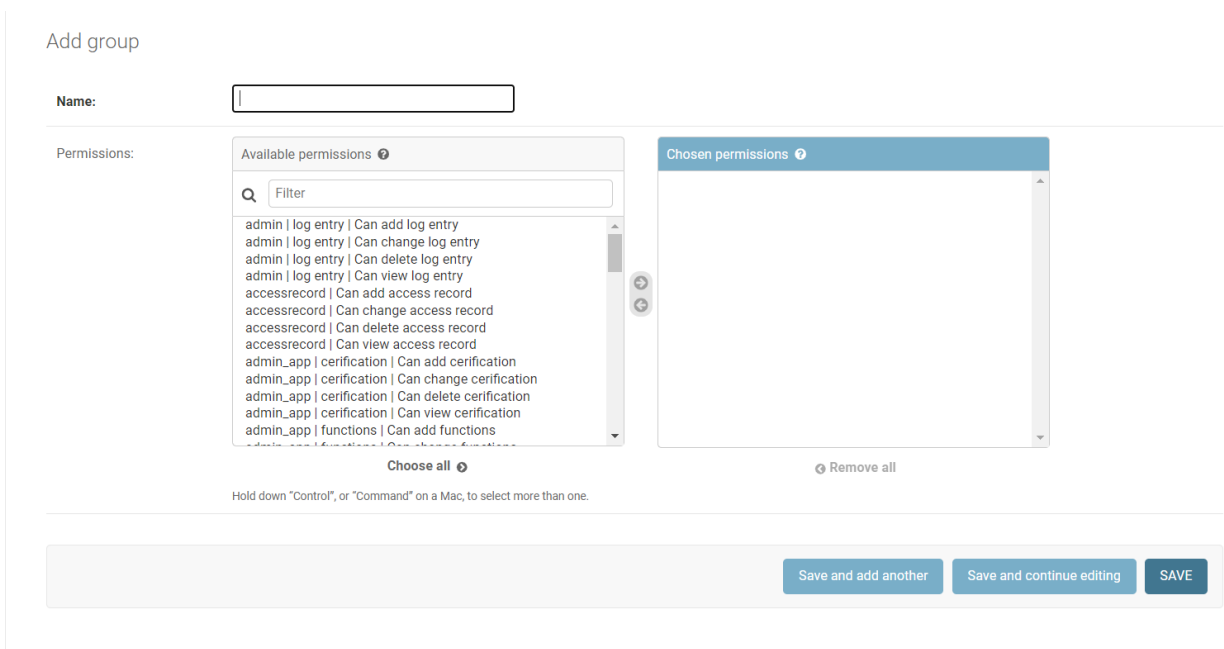


Рисунок 4.3 - Панель контролю доступу

Самі таблиці мають певні поля, які мають бути заповнені для формкування звіту для спеціаліста. В ці поля вноситься інформація про ЗЗІ. Додається опис, назва, виробник, ціна та сайт де є можливість ознайомитись з засобом захисту інформації (рис 4.4).

The screenshot shows an administrator interface with a sidebar menu and a main content area. The sidebar includes sections for 'ADMIN_APP' (with items like Certifications, Functions, Garantylevels, Manufacturers, Meanstypes, and Zasibs) and 'AUTHENTICATION AND AUTHORIZATION' (with Groups and Users). The main content area is titled 'Change zasib' and shows details for 'Канал-201'. The form includes fields for 'Zasib name', 'Zasib url' (with 'Currently' and 'Change' links), 'Zasib number', 'Zasib type number', 'Company maker number', 'Certificate number', and a 'Zasib requirements' text area. A 'HISTORY' button is also present.

Рисунок 4.4 - Приклад заповнення таблиці засобів захисту інформації

Мова використана для створення системи – англійська. В подальшому буде можливість змінювати мову з англійської на українську та навпаки.

Висновки до четвертого розділу

З урахуванням необхідності розробки прототипу програмного забезпечення, функціональності розробленої системи аналізу інформації та внутрішньої бази даних дозволених засобів індивідуального захисту було вирішено створити прототип програмного забезпечення з наступними функціями: Інтерфейс адміністратора для прямої взаємодії з базою даних, вхідними областями фахівця і виходом генерується списку туману .

В ході розробки цього розділу був розроблений прототип програмного забезпечення.

Інтерфейс адміністратора вводить і організовує дані в базу даних і служить джерелом введення для ІАС. Експертна інтерфейс, який приймає рішення про створенн, стане основою для подальшого розвитку призначеного для користувача інтерфейсу ІАС після запровадження відповідних запитів користувачів в критерії відбору. Крім того, текстовий результат цього програмного забезпечення заснований на результатах рекомендованих інструментів для експерта.

Як і зараз, це програмне забезпечення забезпечує зручну навігаційну базу даних для активних носіїв і зручний ручний вибір.

ВИСНОВКИ

При створенні КСЗІ експерт повинен використовувати лише засоби ТЗІ, дозволені для захисту державних інформаційних ресурсів. Переліки, в яких подані відомості про такі ТЗІ, є не надто зручними. Бажаною є реструктуризація та доповнення даних, наведених у цих переліках. Створення ІАС на основі цих переліків значно полегшить експертові прийняття рішення про ТЗІ, необхідні для створення КСЗІ. До цього часу робота ані з реструктуризації переліків, ані зі створення ІАС чи СППР, заснованих на вищезазначених переліках, проведена ніким не була.

Під час цієї роботи був визначений функціонал ІАС, була розроблена структурно-логічна схема роботи ІАС та втілені певні її елементи на практиці. У структурно-логічній схемі чітко визначені джерела вхідних даних, масиви вихідних даних та, у разі незадоволення експерта результуючими рекомендаціями, передбачена можливість змінити вхідні дані як механізм зворотнього зв'язку. ІАС братиме на ввід відомості про роботу підприємства та окремі побажання щодо засобів ТЗІ; на виводі вона видає три рекомендованих переліки засобів ТЗІ, сформовані при різних врахованих факторах.

Практично були втілені наступні елементи: база даних дозволених засобів ТЗІ та прототип програмного забезпечення.

База даних була розроблена з урахуванням потреби інформації експерту на основі даних, що містяться в офіційних переліках дозволених засобів ТЗІ, та інформації, якою можна доповнити ці дані, що стала б цікавою для експерта та брала б участь у прийнятті кінцевого рішення (зокрема, інформація про ринкову вартість продукту). Спочатку була розроблена інфологічна модель бази даних, яка згрупувала інформацію по таблицях та розставила між ними відношення. На її основі була розроблена даталогічна

модель, яка показувала, яким саме чином дані розміщуватимуться на сервері реляційної бази даних. Зв'язки між таблицями даталогічної моделі незначним чином відрізнялися від зв'язків інфологічної моделі, відображаючи реальні потреби розміщувати дані. На заключному етапі база даних була втілена на сервері реляційної бази даних SQLite. Після її наповнення ця база даних стала джерелом вхідних даних як для ІАС, так і для прототипного ПЗ.

Прототип ПЗ був поділений на адміністраторську частину та експертну частину. Адміністраторська частина забезпечує безпосередню взаємодію з базою даних, надаючи зручний інтерфейс для додавання та зміни в ній відповідних записів. Експертна частина надає експертові інтерфейс для вибору конкретних побажань з приводу засобів ТЗІ. У цій частині експерт зможе відфільтрувати перелік за заданими критеріями, обрати бажані засоби та вивести результат. У тому вигляді, який має це ПЗ наразі, воно забезпечує зручну навігацію базою даних та надає експертові можливість відфільтрувати за заданими критеріями перелік, вибрати з відфільтрованого переліку необхідні засоби ТЗІ та вивести перелік обраних засобів у текстовий файл. Цей прототип ПЗ використовуватиметься у подальшому при розробці повноцінної ІАС як базис для вводу користувацьких побажань щодо засобів ТЗІ та виводу рекомендованих переліків ТЗІ.

Подальший напрям досліджень полягає у створенні повноцінної ІАС.

Таким чином, мета роботи, яка полягала у розробці елементів інформаційно-аналітичної системи вибору та аналізу комплексу засобів захисту державних інформаційних ресурсів, досягнута; всі поставлені завдання виконані.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Про захист інформації в інформаційно-телекомунікаційних системах : Закон України від 05.07.1994 р. № 81/94-ВР//ВВР. – 1994. – № 31. – С. 287 [Електронний ресурс]. – Режим доступу: <http://zakon4.rada.gov.ua/laws/show/80/94-вр>.
2. Про Положення про технічний захист інформації в Україні [Електронний ресурс]: указ Президента України від 27.09.1999 № 1229/99. – Режим доступу: <https://zakon.rada.gov.ua/laws/show/1229/99>.
3. Компьютерные информационно-аналитические системы и хранилища данных. Толковый словарь. / А. Г. Додонов, Д. В. Ландэ, С. Р. Коженевский, В. Г. Путятин. – К.: Феникс; ИПРИ НАН Украины, 2013. – 554 с.
4. Що таке база даних? [Електронний ресурс] // Кафедра АПЕПС ТЕФ КПІ – програмна інженерія. – 2019. – Режим доступу до ресурсу: <http://apeps.kpi.ua/shco-take-basa-danykh>.
5. Види програмного забезпечення [Електронний ресурс] // Кафедра АПЕПС ТЕФ КПІ – програмна інженерія. – 2019. – Режим доступу до ресурсу: <http://apeps.kpi.ua/vidi-programnoho-zabezpechenia>.
6. Журавльова І. В. Інформаційно-комунікаційне забезпечення фінансової діяльності / І. В. Журавльова, І. Л. Латишева, О. В. Лебідь. – Харків: Харківський національний економічний університет імені Семена Кузнеця, 2014. – 424 с.
7. Що таке – реляційна база даних? [Електронний ресурс] // EasyCode. – 2010. – Режим доступу до ресурсу: <http://easy-code.com.ua/2010/10/shho-take-relyacijna-baza-danix/>.

8. Гринчуцька С. В. Системи прийняття рішень / С. В. Гринчуцька, Д. В. Дмитрів. – Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. – 130 с.
9. Зибін С. В. Метод імітаційного моделювання функціонування СППР в складі програми інформаційної безпеки / С. В. Зибін. // Сучасний захист інформації. – 2016. – №4. – С. 97–105.
10. Дрейс Ю. О. Система підтримки прийняття рішень процесу управління захисту інформації в інформаційно-телекомунікаційній системі / Ю. О. Дрейс, Л. В. Коваль. // Вісник ЖДТУ. – 2011. – №4 (59). – С. 122–128.
11. Ковтунець В. В. Безпека систем підтримки прийняття рішень / В. В. Ковтунець, О. В. Нестеренко, О. І. Савенков. – К.: ВНЗ «Національна академія управління», 2016. – 190 с.
12. Перелік засобів технічного захисту інформації, дозволених для забезпечення технічного захисту державних інформаційних ресурсів та інформації, вимога щодо захисту якої встановлена законом [Електронний ресурс]. – Режим доступу: http://195.78.68.84/dsszzi/control/uk/publish/article?showHidden=1&art_id=288071&cat_id=44795&ctime=1522825383626.
13. Перелік засобів КЗІ, які мають експертний висновок за результатами державної експертизи у галузі КЗІ [Електронний ресурс]. – Режим доступу: http://www.dsszzi.gov.ua/dsszzi/control/uk/publish/article?art_id=283948&cat_id=72110.
14. Бучик С.С. Актуальність розробки бази даних дозволених засобів технічного захисту інформації / Бучик С. С., Іванченко С. В. // Інформаційні технології та взаємодії : V Міжнародна науково-практична конференція, 20 - 21 листопада 2018 року : тези доп. – К.: Київський нац. ун-т імені Тараса Шевченка, 2018. – С. 258-259.

15. Информационно-аналитическая система и ее создание [Электронный ресурс] // BusinessMan.ru. – 2018. – Режим доступа до ресурсу: <https://businessman.ru/informatsionno-analiticheskaya-sistema-i-ee-sozdanie.html>.
16. Кожевников Д. Г., Лим В. Г. Структура информационно-аналитической системы анализа риска эксплуатации инженерных коммуникаций [Текст] // Современные тенденции технических наук: материалы II Междунар. науч. конф. (г. Уфа, май 2013 г.). — Уфа: Лето, 2013. — С. 48-51.
17. Бучик С.С. Інфологічна модель бази даних технічних засобів інформації, дозволених для забезпечення технічного захисту державних інформаційних ресурсів / Бучик С. С., Іванченко С. В. // Проблеми кібербезпеки інформаційно-телекомунікаційних систем : II Міжнародна науково-практична конференція, 11 –12 квітня 2019 року : тези доп. – К.: Київський нац. ун-т імені Тараса Шевченка, 2019. – С. 287-290.
18. DB-Engines ranking - popularity ranking of database management systems [Электронный ресурс] // DB-Engines. – 2019. – Режим доступа до ресурсу: <https://db-engines.com/en/ranking>.
19. Best Data Types for Currency/Money in MySQL? - Rietta [Электронный ресурс] // The Rietta Blog. – 2012. – Режим доступа до ресурсу: <https://rietta.com/blog/2012/03/03/best-data-types-for-currencymoney-in/>.

ДОДАТКИ

Додаток А

1. Serhii Buchyk, Yaroslav Andrushchenko Searching for a potential criminal using wireless internet networks as one of the targets of state security // Information Technology and Interactions (IT&I–2020) – Київ, 2020, ст. 65
2. Бучик Сергій, Андрущенко Ярослав Development of elements of information-analytical system of selection and analysis of a complex of means of protection of state information resources // IV міжнародної науково-практична конференція «проблеми кібербезпеки інформаційно-телекомунікаційних систем» (pcsits) – Київ, 2021

Додаток Б

Приклад коду створення бази даних ІАС

```

from django.db import models

class Zasib(models.Model):
    zasib_name =
models.CharField(max_length=264,unique=True)
    zasib_url = models.URLField(blank=True)
    zasib_number = models.IntegerField()
    zasib_type_number =
models.PositiveSmallIntegerField(default=0)
    company_maker_number =
models.PositiveSmallIntegerField(default=0)
    zasib_requirements = models.TextField(blank=True,
null=True)
    certificate_number =
models.PositiveSmallIntegerField(default=0)
    functional_profile = models.TextField(blank=True,
null=True)
    garanty_level_number =
models.PositiveSmallIntegerField(default=0)
    official_kzi_category =
models.TextField(blank=True, null=True)
    detailed_description = models.TextField(blank=True,
null=True)
    market_price =
models.PositiveSmallIntegerField(default=0)

    def __str__(self):
        return self.zasib_name

class Meanstype(models.Model):
    means_type_number = models.IntegerField()
    means_type_name =
models.CharField(max_length=264,unique=True)
    information_properies_protected_by_means =
models.TextField(blank=True, null=True)
    means_type_presentatin =
models.TextField(blank=True, null=True)

    def __str__(self):
        return self.means_type_name

class Manufacturer(models.Model):

```

```

    manufacturer_number = models.IntegerField()
    manufacturer_name =
models.CharField(max_length=264, unique=True)
    full_company_adrres_and_name =
models.TextField(blank=True, null=True)
    manufacturer_country =
models.CharField(max_length=50, unique=True)
    product_reputation =
models.CharField(max_length=50, unique=True)
    company_reputation =
models.CharField(max_length=50, unique=True)
    product_reputation_number = models.IntegerField()
    company_reputation_number = models.IntegerField()

```

```

def __str__(self):
    return self.manufacturer_name

```

```

class Functions(models.Model):
    function_number = models.IntegerField()
    abbreviation_functions =
models.CharField(max_length=20, unique=True)
    abbreviation_interpretation =
models.CharField(max_length=100, unique=True)

```

```

def __str__(self):
    return self.function_number

```

```

class Cerification(models.Model):
    certificate_conclusion_inbd_number =
models.IntegerField()
    certificate_or_expert_conclusion =
models.BooleanField(null=False)
    document_number = models.IntegerField()
    date_of_start =
models.DateTimeField(auto_now=False,
auto_now_add=False)
    date_of_end = models.DateTimeField(auto_now=False,
auto_now_add=False)

```

```

def __str__(self):
    return self.certificate_conclusion_inbd_number

```

```

class Garantylevel(models.Model):
    garanty_level_number = models.IntegerField()

```

```
    abbreviation_garanty =  
models.CharField(max_length=100, unique=True)  
    explanation_garanty = models.TextField(blank=True,  
null=True)  
  
def __str__(self):  
    return self.garanty_level_number
```

Додаток В

Приклад коду для побудови рекомендацій для експерту

```

import sqlite3

def create_db_tables():
    conn = sqlite3.connect('zzi.db')
    c = conn.cursor()
    """створення таблиці ЗАСІБ
    zasib_number - Номер засобу (ключ)
    zasib_name - Назва засобу
    zasib_type_number - Номер типу засобу
    company_maker_number - Номер компанії-виробника
    zasib_requirements - Вимоги, яким відповідає
    засіб захисту
    certificate_number - Номер сертифікату
    відповідності чи експертного висновку
    functional_profile - Функціональний профіль
    garanty_level_number - Номер рівня гарантій
    official_kzi_category - Офіційна категорія засобів
    КЗІ, до яких належить засіб
    additional_mean_of_protection_number - Номер
    засобу захисту (або іншого продукту), додатком до якого
    є даний засіб
    detailed_description - Докладніший опис, для чого
    потрібен даний пристрій
    market_price - Ринкова вартість засобу захисту
    """
    c.execute("""CREATE TABLE zasib(
        zasib_number integer,
        zasib_name text,
        zasib_type_number integer,
        company_maker_number integer,
        zasib_requirements text,
        certificate_number integer,
        functional_profile text,
        garanty_level_number integer,
        official_kzi_category text,
        additional_mean_of_protection_number integer,
        detailed_description text,
        market_price integer
    )
    """)
    """створення тблиці ТИП ЗАСОБУ

```

means_type_number - Номер типу засобу (ключ)
 means_type_name - Назва типу засобу
 information_properies_protected_by_means -
 Властивості інформації, що захищає даний тип засобу
 means_type_presentatin - Тип засобу представлений
 апаратним, програмним чи апаратно-програмним комплексом
 """

```
c.execute("""CREATE TABLE means_type(
            means_type_number integer,
            means_type_name text,
            information_properies_protected_by_means text,
            text
        ) """)
```

"""створення таблиці КОМПАНІЯ-ВИРОБНИК

company_maker_number - Номер компанії-виробника
 (ключ)

company_maker_name - Назва компанії-виробника

full_company_adrres_name - Повна назва компанії-
 виробника та її адреса

company_maker_country - Країна походження
 компанії-виробника

product_reputation - Репутація продуктів виробника
 (текстове пояснення)

company_reputation - Репутація компанії-виробника
 (текстове пояснення)

product_reputation_int - Репутація продуктів
 виробника (число)

company_reputation_int - Репутація компанії-
 виробника (число)

"""

```
c.execute("""CREATE TABLE manufacturer(
            company_maker_number integer,
            compay_maker_name text,
            full_company_adrres_name text,
            company_maker_country text,
            product_reputation text,
            company_reputation text,
            product_reputation_int integer,
            company_reputation_int integer
        ) """)
```

"""створення таблиці ФУНКЦІЇ

function_number - Репутація компанії-виробника
 (число)

"""

```
c.execute("""CREATE TABLE functions(
            function_number int,
            abbreviation_functions text,
            abbreviation_interpretation text
        ) """)
```

```
"""створення таблиці СЕРТИФІКАТ-ВИСНОВОК
certificate_conclusion_inbd_number - Номер (у БД)
сертифікату чи експертного висновку (ключ)
certificate_or_expert_conclusion - Сертифікат чи
експертний висновок
document_number- Номер документу
date_of_start - Дата початку дії
date_of_end - Дата закінчення дії
"""
```

```
c.execute("""CREATE TABLE cerification(
            certificate_conclusion_inbd number integer,
            certificate_or_expert_conclusion
boolean,
            document_number integer,
            date_of_start text,
            date_of_end text
        ) """)
```

```
"""створення таблиці РІВЕНЬ ГАРАНТІЇ
garanty_level_number - Номер рівня гарантій (ключ)
abbreviation_garanty - Аббревіатура
explanation_garanty - Пояснення
"""
```

```
c.execute("""CREATE TABLE garanty_level(
            garanty_level_number integer,
            abbreviation_garanty text,
            explanation_garanty text
        ) """)
```

```
conn.commit()
conn.close()
create_db_tabels()
```

Додаток Г

Програма керуванням та фільтрації таблиць за прикладом таблиці “Засоби”

```
import sqlite3
```

```
"""
```

```
Набір функцій для керування таблицею ЗАСІВ
```

```
"""
```

```
def show_all():
```

```
    """Функція виводить всю інформацію з таблицьки"""
```

```
    conn = sqlite3.connect('zzi.db')
```

```
    c = conn.cursor()
```

```
    c.execute("SELECT * FROM zasib")
```

```
    items = c.fetchall()
```

```
    for item in items:
```

```
        print(item)
```

```
    conn.commit()
```

```
    conn.close()
```

```
def add_record(zasib_number, zasib_name,
               zasib_type_number, company_maker_number,
               zasib_requirements,
               certificate_number, functional_profile='',
               garanty_level_number=0,
               official_kzi_category,
               additional_mean_of_protection_number=0,
               detailed_description,
               market_price=0):
```

```
    """Функція додає запис в таблицьку ЗАСІВ"""
```

```
    conn = sqlite3.connect('zzi.db')
```

```
    c = conn.cursor()
```

```
    c.execute("INSERT INTO zasib VALUES
(?,?,?,?,?,?,?,?,?,?,?,?)", (zasib_number, zasib_name,
                               zasib_type_number, company_maker_number,
                               garanty_level_number,
                               official_kzi_category,
                               additional_mean_of_protection_number,
                               detailed_description,
                               market_price))
```

```
    conn.commit()
```

```
    conn.close()
```

```

def add_many_records(records_list):
    """Функція додає перелік записів до таблиці
    ЗАСІВ"""

    conn = sqlite3.connect('zzi.db')

    c = conn.cursor()
    c.executemany("INSERT INTO zasib VALUES
    (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)", (records_list))

    conn.commit()
    conn.close()

def delete_record(zasib_number):
    """Функція видаляє запис з таблиці ЗАСІВ згідно
    номеру засобу"""
    conn = sqlite3.connect('zzi.db')

    c = conn.cursor()
    c.execute("DELETE FROM zasib WHERE zasib_number =
    (?)", zasib_number)

    conn.commit()
    conn.close()

def search_record(column_name, record):
    """Пошук запису по назві колонки та запису в цій
    колонці"""
    conn = sqlite3.connect('zzi.db')

    c = conn.cursor()
    c.execute("SELECT * FROM zasib WHERE {} =
    (?)".format(column_name), (record,))

    items = c.fetchall()
    for item in items:
        print(item)

    conn.commit()
    conn.close()

```

```

1.
2.<!DOCTYPE html>
3.<html lang="" dir="ltr">
4.  <head>
5.    <meta charset="utf-8">
6.    <title>Головна сторінка</title>
7.    <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap
      /4.1.3/css/bootstrap.min.css"
8.    integrity="sha384-
      MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkF
      OJwJ8ERdknLPMO" crossorigin="anonymous">
9.    <!-- <link rel="stylesheet"
      href="https://stackpath.bootstrapcdn.com/bootstrap
      /4.1.3/css/bootstrap.min.css"
10.     integrity="sha384-
      MCw98/SFnGE8fJT3GXwEOngsV7Zt27NXFoaoApmYm81iuXoPkF
      OJwJ8ERdknLPMO" crossorigin="anonymous"> -->
11.     <style media="screen">
12.
13.     </style>
14.   </head>
15.
16.
17.   <body>
18.     <nav class="navbar navbar-expand-md navbar-
      dark bg-dark">
19.       <div class="navbar-collapse collapse w-
      100 order-1 order-md-0 dual-collapse2">
20.         <ul class="navbar-nav mr-auto">
21.           <li class="nav-item active">
22.             <a class="nav-link"
      href="#">Початкова сторінка</a>
23.           </li>
24.           <li class="nav-item">
25.             <a class="nav-link"
      href="infoPage.html">Інформація</a>
26.           </li>
27.         </ul>
28.       </div>
29.       <div class="navbar-collapse collapse w-100
      order-3 dual-collapse2">
30.         <ul class="navbar-nav ml-auto">
31.           <li class="nav-item">

```

```

32.         <a class="nav-link" id="login"
    href="loginPage.html">Увійти</a>
33.     </li>
34. </nav>
35.
36.     <div class="container">
37.         <div class="jumbotron">
38.             <h1>Вас вітає Інформаційно-Аналітична
    Система</h1>
39.             <h2>Будь-ласка, оберіть свою роль
    згідно цілей використання ситеми</h2>
40.         </div>
41.
42.         <div class="d-grid gap-2 col-12 btn-
    group-vertical mx-auto">
43.             <button class="btn btn-dark"
    type="button"><a
    href="http://127.0.0.1:8000/admin">Панель
    Адміністратора</a></button>
44.             <button class="btn btn-dark"
    type="button" href="userPage.html">Панель
    Користувача</button>
45.
46.         </div>
47.     </div>
48.
49.
50. </body>
51. </html>

```

```

1.  body{
2.  font-family: 'Montserrat', sans-serif;
3.  color: white;
4.  border: 60px solid #2EC4B6;
5.  border-top: 0px;
6.  background-color: #040c47;
7.  margin: 0px 0px 0px 0px
8. }
9. button {
10.     background-color: #4CAF50; /* Green */
11.     border: 1px solid green;
12.     color: white;
13.     padding: 15px 32px;
14.     text-align: center;
15.     text-decoration: none;

```

```
16.     font-size: 16px;
17.     cursor: pointer;
18.     width: 150px;
19.     display: block;
20.     text-align: center;
21. }
22.
23.
24. #sub{
25.     border-radius: 25px;
26.     font-size: 1.2em;
27.     background: orange;
28.     height: 80px;
29.     width: 150px;
30.     font-family: "Montserrat";
31.     margin-bottom: 4px;
32. }
33. label{
34.     font-size: 1em;
35. }
36.
37. h1{
38.     text-align: center;
39. }
40.
41. h2{
42.     text-align: center;
43.     margin-bottom: 50px;
44. }
45.
46. form{
47.     text-align: center;
48. }
49.
50. p{
51.     text-align: center;
52.     /* margin: 100px 400px 30px 400px; */
53.     padding-top: 5%;
54.     padding-left: 20%;
55.     padding-right: 20%;
56. }
```