

**Методичні вказівки
до виконання курсової роботи з дисципліни
"Організація баз даних і знань"
для студентів напряму підготовки 6.050103
"Програмна інженерія"
денної та заочної форм навчання**

Міністерство освіти і науки України
Вінницький національний технічний університет

Методичні вказівки
до виконання курсової роботи з дисципліни
"Організація баз даних і знань"
для студентів напрямку підготовки 6.050103
"Програмна інженерія"
денної та заочної форм навчання

Вінниця
ВНТУ
2010

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 4 від 17.12. 2009 р.)

Рецензенти:

В. І. Месюра, кандидат технічних наук, доцент

В. М. Лисогор, доктор технічних наук, професор

Методичні вказівки до виконання курсової роботи з дисципліни "Організація баз даних і знань" для студентів напряму підготовки 6.050103 "Програмна інженерія" денної та заочної форм навчання / Уклад. О. Н. Романюк, А. В. Денисюк. – Вінниця: ВНТУ, 2010. – 51 с.

У методичних вказівках визначаються завдання і методи розробки курсової роботи "Організація баз даних і знань", вимоги щодо виконання курсових робіт. Для полегшення самостійного вивчення дисципліни наводиться список рекомендованої літератури.

Призначений для студентів спеціальності "Програмна інженерія" денної та заочної форм навчання.

ЗМІСТ

ВСТУП.....	4
1 ЗМІСТ ТА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ	5
2 ТЕМАТИКА КУРСОВОГО ПРОЕКТУВАННЯ	6
3 МЕТОДИКА ВИКОНАННЯ РОЗДІЛІВ ПРОЕКТУ	6
3.1 Завдання на курсову роботу	6
3.2 Аналіз сучасного розвитку баз даних	6
3.3 Змістовне формування задачі	7
3.4 Постановка задачі та аналіз предметної області	7
3.5 Розробка ER- моделі предметної області.....	7
3.6 Розробка універсального відношення	7
3.7 Проектування нормалізованих відношень.....	7
3.8 Виявлення початкових відношень за методом " суть -зв'язок"	7
3.9 Нормалізація відношень методом декомпозиції.....	8
3.10 Оцінка спроектованих НФБК-відношень	8
3.11 Аналіз реалізованих запитів	8
3.12 Розробка вихідних форм.....	8
3.13 Розробка структури СУБД та схеми даних.....	8
3.14 Опис типів полів розроблених таблиць	8
3.15 Розробка інструкції користувача СУБД.....	8
3.16 Висновки	8
4 СТРОКИ ВИКОНАННЯ КУРСОВОЇ РОБОТИ.....	9
5 ПОРЯДОК ЗАХИСТУ КУРСОВОЇ РОБОТИ	9
6 ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ	10
7 КОРОТКІ ТЕОРИТИЧНІ ПОЛОЖЕННЯ	10
7.1 Аналіз предметної області та постановка задачі.....	10
7.2 Розробка універсального відношення	12
7.3 Розробка ER-моделі предметної області.....	14
7.4 ПРОЕКТУВАННЯ НОРМАЛІЗОВАНИХ ВІДНОШЕНЬ.....	21
7.5 Отримання попередніх відношень за методом “Суть – зв’язок”	25
7.6 Нормалізація відношень методом декомпозиції.....	28
7.7 Оцінка спроектованих НФБК-відношень	30
8 РЕАЛІЗАЦІЯ ЗАПИТІВ І ВИХІДНИХ ФОРМ.....	32
8.1 Аналіз реалізованих БД запитів.....	32
8.2 Розробка вихідних форм.....	33
9 ЗРАЗОК ОФОРМЛЕННЯ ДОДАТКІВ	34
ЛІТЕРАТУРА.....	45
Додаток А. Титульний аркуш	47
Додаток Б. Сторінка індивідуального завдання.....	48
Додаток В. Оформлення тексту пояснювальної записки.....	50

ВСТУП

В умовах різкого зростання інформаційних потоків перед людством повстала надзвичайно важлива задача, яка полягає в необхідності швидкого отримання із великого обсягу інформації потрібних відомостей (даних). Сьогодні це можливо завдяки використанню баз даних, які суттєво змінили характер діяльності людини. Поява комп'ютерних баз даних з їх можливостями щодо забезпечення швидкого доступу до потрібної інформації, в змозі змінити природу еволюції людства і має не менше значення, чим винахід друкарського верстата.

Навчальним планом з курсу "Організація баз даних і знань" передбачається виконання курсової роботи, що сприяє більш глибокому вивченню основних розділів курсу, детальному ознайомленню з основними етапами формування бази даних і поглибленому оволодінню методикою їх проектування. Крім того, у процесі виконання роботи студенти одержують навички програмування мовою СУБД Fox Pro (або Microsoft Office Access 2003).

1 ЗМІСТ ТА ОФОРМЛЕННЯ КУРСОВОЇ РОБОТИ

Зміст курсової роботи повністю визначається завданням, яке видається кожному студенту. Курсове проектування включає декілька послідовних етапів, які пов'язані з аналізом предметної області, проектуванням нормалізованих відношень, розробкою вихідних форм та структури файлів, розробкою алгоритмів та програм мовою обраної СУБД. Графічна частина виконується згідно з вимогами ЕСКД та ЕСПД.

Пояснювальна записка складається з	
1. Аналізу сучасного розвитку баз даних.	1 с.
2. Змістовного формування задачі.	1-2 с.
3. Аналізу предметної області.	1 с.
4. Розробки універсального відношення.	2-3 с.
5. Розробка ER-моделі предметної області.	4-5 с.
6. Проектування нормалізованих відношень.	2-3 с.
6.1. Розробка початкових відношень за методом "суть-зв'язок".	5-6 с.
6.2. Нормалізація відношень методом декомпозиції.	4-5 с.
6.3. Оцінка спроектованих НФБК відношень	1-2 с.
7. Реалізація запитів і вихідних форм	1-2 с.
7.1. Аналіз БД запитів	3-4 с.
7.2. Розробка вихідних форм	3-4 с.
ВИСНОВКИ	1-2 с.
Список використаної літератури	1-2 с.
Додаток А. Форми.	1-2 с.
Додаток Б. Таблиці.	1-2 с.
Додаток В. Запити.	1-2 с.
Додаток Г. Звіти.	1-2 с.
Додаток Д. Схема даних.	1-2 с.

Титульний аркуш пояснювальної записки оформляється за зразком, що поданий в додатку .

Графічна частина курсової роботи (1 аркуш формату А1) повинна складатися з:

- 1) ER-схеми предметної області; діаграми, яка ілюструє процес нормалізації відношень методом декомпозиції;
- 2) схеми даних СУБД, роздруківки шаблонів таблиць запитів і звітів.

2 ТЕМАТИКА КУРСОВОГО ПРОЕКТУВАННЯ

Тема курсової роботи визначається номером завдання $N=k+p$, де k -число, що видається викладачем кожній групі студентів; p - порядковий номер студента в групі.

Дозволяється самостійний вибір студентом теми курсової роботи за узгодженістю з викладачем.

3 МЕТОДИКА ВИКОНАННЯ РОЗДІЛІВ ПРОЕКТУ

3.1 Завдання на курсову роботу

В завданні на курсову роботу необхідно вказати номер варіанта, сформулювати завдання, вибрати мову програмування.

Розробка завдання повинна задовольняти такі мінімальні вимоги:

- ступінь універсального відношення, не менше 12;
 - потужність універсального відношення, не менше 12;
 - кількість "сутей" ER-діаграми, не менше 4;
 - кількість попередніх відношень, не менше 4;
 - форма нормалізації первинних відношень, не менше 3;
 - кількість вихідних форм, не менше 5;
 - кількість запитів, що реалізуються, не менше 5;
 - вхідна мова програмування, Fox Pro + (або Microsoft Office Access 2003 за узгодженістю з викладачем);
- База даних повинна забезпечувати ;
- роботу користувача у режимі кольорового меню;
 - введення, видалення, оновлення інформації;
 - видачу відповідей на запити щодо вибору користувача на екран дисплея або ж пристрій друку;
 - режим підказки для роботи користувача;
 - збереження файлів;
 - контроль пріоритетів доступу до окремих блоків даних шляхом накладання пароля .

3.2 Аналіз сучасного розвитку баз даних

Цей підрозділ є обов'язковим та передбачає посилання до відомих вітчизняних аналогів, враховуючи тенденції розвитку та сучасний стан розвитку баз даних.

Порівняльний аналіз та обґрунтування теми проекту повинні здійснюватись на рівні інженерного мислення із всебічним використанням сучасних досягнень науки та техніки.

3.3 Змістове формування задачі

У цьому підрозділі необхідно вказати :

- область застосування комп'ютерних баз даних, основні переваги, які вони надають, а також їх місце та значення в сучасних умовах;
- ввести основні позначення, котрі будуть використовуватися в процесі виконання курсової роботи;
- сформулювати основні цілі проектування баз даних.

3.4 Постановка задачі та аналіз предметної області

Вказується призначення розробленої бази даних і на основі аналізу предметної області формуються: основні вимоги до даних, що зберігаються у базі; функції, які виконує база даних; основні запити на видачу інформації, основні вимоги до апаратного забезпечення .

3.5 Розробка ER- моделі предметної області

Наводяться короткі теоретичні положення про відхід до проектування баз даних, що ґрунтується на поняттях "суть-зв'язок". Визначаються :

- основна суть, яка адекватно відображає досліджувану предметну область;
- атрибути, які однозначно ідентифікують суть, котра буде використовуватись як ключ відповідного відношення;
- зв'язки, які мають місце між виявленими сутями;
- характеристики кожного зв'язку.

Результати аналізу зводяться в таблицю. Для кожного типу зв'язку необхідно навести відповідну ER-діаграму.

За результатами аналізу здійснюється побудова ER-моделі.

3.6 Розробка універсального відношення

Здійснюється вибір набору атрибутів, які повністю характеризують предметну область. Атрибути разом з описом накладених на них обмежень зводяться в таблицю. Наводиться схема універсального відношення.

3.7 Проектування нормалізованих відношень

Наводяться основні теоретичні положення про нормальні форми 1НФ...5НФ та методи їх нормалізації.

3.8 Виявлення початкових відношень за методом " суть -зв'язок"

На основі ступенів відношень з діаграм ER-типу формується сукупність

попередніх відношень, які зводяться в таблицю. В результаті обробки наведених у табличній формі відношень формуються кінцеві відношення.

3.9 Нормалізація відношень методом декомпозиції

Наводяться основні теоретичні положення методу декомпозиції. На основі аналізу атрибутів універсального відношення виявляються функціональні залежності, що мають місце у відношенні.

З використанням діаграм функціональних залежностей здійснюється послідовна декомпозиція універсального відношення на відношення, що знаходяться в нормальній формі Бойса-Кодда (НФБК).

3.10 Оцінка спроектованих НФБК-відношень

Здійснюється аналіз правильності отриманих відношень.

3.11 Аналіз реалізованих запитів

Описуються запити, які повинні реалізуватися у базі даних. Коротко описується порядок реалізації кожного запиту та відношення, які при цьому використовуються.

3.12 Розробка вихідних форм

Наводиться опис вихідних форм, в котрих використовується інформація, що виводиться за запитам.

3.13 Розробка структури СУБД та схеми даних

Наводиться схема даних створеної СУБД і коротко описуються зв'язки.

3.14 Опис типів полів розроблених таблиць

Коротко описується склад полів таблиць створеної СУБД. Робота з формами.

3.15 Розробка інструкції користувача СУБД

Наводиться короткий опис навігації в СУБД та особливості інтерфейсу під час роботи з базою даних.

3.16 Висновки

Наводяться основні результати проектування бази даних.

4 СТРОКИ ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Рекомендується такий графік виконання курсової роботи при тривалості семестру - 14 навчальних тижнів.

Таблиця 1 – Графік виконання курсової роботи

Розділ	Зміст розділу	% від об'єму КР	Строк виконання (тиждень)
1	Змістовне формування задачі. Розробка технічного завдання	7	1-2
2	Розробка універсального відношення	2	3
3	Розробка ER-моделі	5	3-4
4	Формування початкових відношень за методом "суть - зв'язок"	3	3-4
5	Нормалізація відношень методом декомпозиції	3	3-4
6	Оцінка спроектованих НФБК відношень	5	4
7	Аналіз запитів до БД, що реалізуються	4	5
8	Опис вихідних форм	3	5
9	Обґрунтування запропонованої структури бази даних і схеми даних	5	5
10	Опис складу бази даних та її призначення	13	5-6
11	Розробка структури бази даних і схеми даних	15	7-8
12	Розробка та налагодження бази даних	15	9- 10
13	Розробка інструкції користувача	10	11
14	Оформлення курсової роботи	10	12
15	Захист курсової роботи		13- 14

5 ПОРЯДОК ЗАХИСТУ КУРСОВОЇ РОБОТИ

Готовність до захисту курсової роботи визначає керівник. Якщо робота виконана в повному обсязі і немає принципових помилок, керівник допускає студента до захисту. В іншому випадку робота повертається студенту на доопрацювання. Після позитивного висновку про готовність курсової роботи студент повинен захистити її перед комісією в складі двох викладачів, виділених кафедрою.

Під час захисту студент коротко (5-7 хвилин) доповідає про основні

результати роботи і відповідає на запитання членів комісії за змістом роботи.

6 ЗАВДАННЯ НА КУРСОВЕ ПРОЕКТУВАННЯ

Теми курсових робіт наведені в кінці методичних вказівок. Позитивно оцінюється самостійний вибір студентом (за узгодженням з викладачем) теми курсової роботи. Самостійний вибір предметної області, у якій доцільно використовувати бази даних, дозволяє скласти враження про рівень творчої активності студента, його вміння здійснити попередній аналіз предметної області.

7 КОРОТКІ ТЕОРИТИЧНІ ПОЛОЖЕННЯ

З метою полегшення виконання курсової роботи наводяться приклади виконання її окремих розділів, але забороняється використовувати ці розділи як електронний шаблон курсової роботи.

7.1 Аналіз предметної області та постановка задачі

Першим етапом проектування бази даних будь-якого типу є аналіз предметної області, що закінчується побудовою інформаційної структури (концептуальної схеми) [1]. На даному етапі аналізуються запити користувачів, вибираються інформаційні об'єкти та їх характеристики і на основі проведеного аналізу формується структура предметної області, яка не залежить від програмного та технічного середовища, в якому буде реалізуватися база даних.

Аналіз предметної області доцільно розбити на три фази:

- аналіз концептуальних вимог та інформаційних потреб;
- виявлення інформаційних об'єктів та зв'язків між ними;
- побудова концептуальної моделі предметної області та проектування концептуальної схеми бази даних.

На етапі аналізу концептуальних вимог та інформаційних потреб необхідно вирішити такі задачі:

- аналіз вимог користувача до бази даних (концептуальних вимог);
- виявлення задач, що мають місце при обробці інформації, яка повинна бути подана у базі даних;
- виявлення перспективних задач;
- документування результатів аналізу.

Вимогами користувачів до бази даних, що розробляється, є, в загальному випадку, список запитів з вказанням їх інтенсивності та об'ємів даних. Ці вказівки опрацьовуються в діалозі з майбутнім користувачем бази даних. Тут же з'ясовуються вимоги до введення, відновлення та корегування інформації. Вимоги користувачів уточнюються та доповнюються при аналізі

перспективних задач, що мають місце.

Розглянемо приклад створення бази даних для контролю оплати по кредитах.

База даних з предметної області “Контроль оплати по кредитах” призначена для використання в банківських установах. Тому слід врахувати, що головною її функцією є надання чіткої інформації про стан погашення наданих кредитів, щоб банківська установа могла негайно прийняти відповідні рішення, адже велика не погашена сума може призвести до банкрутства. Також однією з вимог до даної бази даних є зберігання списків боржників, щоб в подальшому не надавати їм кредиту. Також база даних повинна автоматично розраховувати суму повернення згідно з чинною кредитною ставкою, визначати повернену суму, і зберігати усі виконані клієнтом фінансові операції.

Згідно з вищесказаними вимогами в базу даних слід включити: дані про клієнтів; види кредитів; інформацію про оформлення кредиту; а також дані про погашення кредиту.

База даних повинна виконувати такі функції.

1. Введення і зберігання даних про клієнтів:
 - прізвище, ім'я, та по батькові;
 - ідентифікаційний код;
 - адреса;
 - рік народження.
2. Введення і зберігання видів кредитів, які клієнти можуть отримати:
 - назва кредиту;
 - максимальна сума;
 - річні проценти.
3. Оформлення кредиту:
 - код кредиту;
 - сума кредиту;
 - дата надання;
 - дата завершення.
4. Введення і зберігання інформації про погашення кредиту:
 - сума;
 - дата;
 - час.
5. Виявлення боржників, що не повернули кредит до зазначеної дати:
 - повернення.
6. Пошук клієнтів за адресою.
7. Формування списку клієнтів, які вже погасили кредит і можуть знову отримувати кредит.
8. Формування інформації про кредитну діяльність клієнтів.

База даних “Контроль оплати по кредитах” розробляється в середовищі Microsoft Office Access 2003, і може бути виконана на комп’ютері з середовищем Microsoft Office Access. Введення інформації здійснюється за допомогою клавіатури через форми в таблиці де інформації виводиться на екран комп’ютера. Також існує можливість виведення інформації на принтер.

7.2 Розробка універсального відношення

Друга фаза аналізу предметної області складається з: вибору інформаційних об’єктів, задання необхідних властивостей для кожного об’єкта, виявлення зв’язків між об’єктами, виявлення обмежень, що накладаються на інформаційні об’єкти, типи зв’язків між ними, характеристики інформаційних об’єктів.

При виборі інформаційних об’єктів бажано намагатися відповісти на такі питання.

1. На які класи можна розбити дані, що підлягають зберіганню у базі даних?

2. Яке ім’я можна присвоїти кожному класу даних?

3. Які найбільш цікаві характеристики (з точки зору користувача) кожного класу даних можна виділити?

4. Які імена можна присвоїти вибраним наборам характеристик?

Виділення інформаційних об’єктів - процес ітеративний. Він здійснюється на основі аналізу інформаційних потоків та інтерв’ювання споживачів. Характеристики інформаційних об’єктів визначаються тими ж методами.

Введемо ряд позначень, котрі будуть використовуватися у ході подальшого викладення матеріалу.

R - є відношення над множинами D_1, D_2, \dots, D_n якщо воно являє собою множину упорядкованих n -кортежів вигляду d_1, d_2, \dots, d_n . D_1, D_2, \dots, D_n - називаються доменами відношення R .

Відношення може бути подане у вигляді файла або таблиці, стовпці котрих - елементи доменів, а рядки - кортежі. Кожен кортеж відображає один екземпляр інформаційного об’єкта. Імена стовпців (поле запису) називаються атрибутами, а індивідуальні значення елементів - значеннями атрибутів. Кожен атрибут відображає відповідну характеристику інформаційного об’єкта. Число стовпців у відношенні називається ступенем відношення, а число кортежів - потужністю відношення. У процесі експлуатації бази даних ступінь відношення змінюється значно рідше, чим його потужність.

Реляційна база даних являє собою сукупність відношень, що зберігають всю необхідну інформацію. Кожне відношення зберігається в

окремому файлі з дуже простою структурою, оскільки всі записи файла мають однаковий формат.

Атрибут або набір атрибутів, котрий можна використовувати для однозначної ідентифікації конкретного кортежу, називається початковим ключем (у випадку набору атрибутів - складений ключ). Можливі випадки, коли відношення може вміщувати декілька унікальних ключів. Тоді один з них вибирається як головний початковий, а інші отримують назву початкових ключів.

Атрибути, що представляють копії ключів інших відношень, називаються зовнішніми ключами.

Атрибут або набір атрибутів, що використовуються для більш швидкого пошуку, називається другорядним індексом.

УНІВЕРСАЛЬНИМ ВІДНОШЕННЯМ називається відношення, що вміщує в себе всі атрибути, котрі будуть використовуватися у базі даних. Для невеликих баз даних універсальне відношення може служити відправною точкою при їх проектуванні.

Розглянемо порядок створення універсального відношення при розробці бази даних для контролю оплати по кредитах. Виходячи з виконаного в минулому розділі аналізу предметної області, в універсальне відношення потрібно включити атрибути, що описують таку інформацію про об'єкт:

особа, види кредитів, кредит, щомісячна оплата.

Перерахуємо атрибути вищеназваних об'єктів:

Особа (ПІБ, ідентифікаційний код, адреса, рік народження).

Види кредитів (назва кредиту, максимальна сума, річні проценти).

Кредит (код кредиту, сума кредиту, дата надання, дата завершення).

Щомісячна оплата (сума, дата, час).

Для деяких характеристик об'єктів необхідно визначити обмеження. Логічні обмеження, що накладаються на дані, називаються обмеженням цілісності. Обмеження цілісності це така властивість, котра задається для деякого інформаційного об'єкта або його характеристик і котра повинна зберігатися для кожного стану.

Обмеження цілісності визначимо для атрибута *рік народження* відношення *особа*. Як відомо кредити не надаються особам віком менше 18 років. Також *дата завершення* відношення *кредит* повинна бути більша за дату оформлення кредиту, що є цілком очевидним.

В таблиці 2 перерахуємо атрибути для універсального відношення.

Таблиця 2 – Початковий перелік атрибутів, для формування універсального відношення бази даних

Назва атрибута	Ім'я поля	Коментарій
ПІБ	ПІБ	Прізвище, ім'я, та по батькові клієнта
ідентифікаційний код	ідентифікаційний код	Ідентифікаційний код клієнта
адреса	адреса	Адреса клієнта
рік народження	рік народження	Рік народження клієнта
назва кредиту	назва кредиту	Назва кредиту, який клієнт може взяти
максимальна сума	максимальна сума	Максимальна сума кредиту
річні проценти	річні проценти	Проценти, під які позичаються кредити
код кредиту	код кредиту	Унікальний
сума кредиту	сума кредиту	Сума, яку клієнт позичає
дата надання	дата надання	Дата надання кредиту
дата завершення	дата завершення	Кінцева дата погашення кредиту
сума	сума	Сума погашення
дата	дата	Дата погашення
час	час	Час погашення

Оскільки всі перераховані в таблиці атрибути є незалежними, тобто значення одних з них не можуть бути обчислені за значеннями інших, то всі вони можуть бути включеними в склад універсального відношення.

R (ПІБ, ідентифікаційний код, адреса, рік народження, назва кредиту, максимальна сума, річні проценти, код кредиту, сума кредиту, дата надання, дата завершення, сума, дата, час).

7.3 Розробка ER-моделі предметної області

Заключна фаза аналізу предметної області складається з розробки її інформаційної структури (або концептуальної схеми).

В звичайних випадках для побудови концептуальної схеми використовуються традиційні методи агрегації та узагальнення. При агрегації декілька інформаційних об'єктів (елементів даних) об'єднуються в один відповідно до семантичних зв'язків між об'єктами. Наприклад, літак типу А перевозить вантаж з пункту відправлення в пункт

надходження. Методом агрегації можна створити інформаційний об'єкт (суть) РЕЙС з атрибутами: тип літака, пункт відправлення, пункт надходження, рейс літака.

При узагальненні інформаційні об'єкти (елементи даних) об'єднуються в родовий об'єкт. Наприклад, інформаційні об'єкти: автобус, автомобіль, літак, пароплав - можуть бути об'єднані в родовий об'єкт "транспортні засоби".

Концептуальна модель застосовується для структурування предметної області з урахуванням не тільки інформаційних інтересів користувачів системи, але і інформаційного споживання самої предметної області.

Концептуальна модель дозволяє чітко розглянути окремі елементи предметної області. При цьому детальність відображення предметної області залежить від вибраної моделі. Модель з мінімальними можливостями повинна забезпечувати здатність задання даних та їх взаємозв'язок. Відповідно, семантична потужність концептуальної моделі збільшується із зростанням додаткового числа характеристик, котрі вона дозволяє визначити.

При виборі моделі для концептуального проектування бажано враховувати ті обставини, що будь-яким моделям властиві деякі обмеження, тому пошук ідеальної моделі, що повністю відображає реальний світ, дуже проблематичний. Вибір моделі диктується насамперед характером предметної області і вимогами, що ставляться перед базою даних. Іншою важливою обставиною є незалежність концептуальної моделі від СУБД, котра повинна бути вибрана після побудови концептуальної схеми.

В курсовій роботі для подання структури і обмежень реального світу використовуються моделі типу "суть-зв'язок" або ER-моделі, котрі були введені у практику П. Ченом [2, 3, 8].

Основними конструкціями ER-моделі є суті і зв'язки [5,7,13,14].

Під суттю розуміють основний зміст того явища, процесу або об'єкта, про який збирають інформацію в базі даних. Як суть може виступати місце, річ, особа і так далі. Суть визначається як деякий об'єкт, який цікавий користувачу. Суть, як правило виступає іменником.

Розрізняють тип суті та екземпляр суті. Під типом суті розуміють набір однорідних об'єктів, що виступають як ціле. Поняття "екземпляр суті" відноситься до конкретного об'єкта. Наприклад: тип суті – службовець, екземпляр суті - Коваленко Василь Васильович. Суть прийнято зазначати поіменованими характеристиками, які мають назву атрибутів. Назва атрибута повинна бути унікальною для кожної суті, хоча вона може повторюватися для різноманітних типів суті.

Наприклад:

тип суті – службовець

атрибути - прізвище, ім'я та по-батькові, адреса, рік народження і т. д.

тип суті - хворий

атрибути - прізвище, ім'я, по батькові, адреса, діагноз і т. д.

Розрізняють тип атрибута та екземпляр атрибута. Тип атрибута задає властивість суті, а екземпляри атрибута відображають різноманітні значення цієї властивості.

Наприклад:

тип атрибута - дата народження

екземпляри атрибута - 20.05.1941, 5.06.1958 і т. д.

Надалі при використанні терміну "суть" будемо розуміти тип суті, а терміну "атрибут" - тип атрибута.

Зазначимо, що абсолютна відміна між суттю і атрибутом відсутня. Атрибут є типом тільки в зв'язку з обраною суттю, а в іншому контексті він сам може виступати як суть. Наприклад, для заводу, що виготовляє автомобілі, колір фарби є лише одним з атрибутів суті автомобіля. Для фабрики, що випускає фарби, колір є суттю.

Між сутями (інформаційними об'єктами), а також між суттю і її атрибутами з'являються деякі асоціації, що називаються зв'язками. При цьому зв'язки можуть бути різних властивостей, характеру та вибірності. Звичайно зв'язок виражається дієсловом. Зв'язки як і суті іменуються, при цьому екземпляр кожного окремого зв'язку специфікується лінією між тими двома екземплярами суті, котрі цей зв'язок з'єднують. Атрибут (або набір атрибутів), що використовується для ідентифікації суті, називається ключем суті.

Кожен екземпляр зв'язку визначається набором ключів суті, що з'єднуються цим зв'язком.

Суті можуть з'єднуватися зв'язками різної ступені. Наприклад, ступінь зв'язку 1:1 свідчить про те, що один екземпляр однієї суті може бути зв'язаний не більше, ніж з одним екземпляром іншої суті. Відповідно, ступінь зв'язку 1:n означає, що один екземпляр однієї суті може бути зв'язаний з декількома екземплярами іншої суті, але не навпаки. Накінець, ступінь зв'язку m:n показує, що кожен з екземплярів будь-якої з сутей може бути зв'язаний з декількома екземплярами іншої суті.

Зв'язки можуть подаватися різними способами, з яких ми будемо використовувати тільки два: діаграма ER-екземплярів і діаграма ER-типів. Діаграми ER-екземплярів відображають зв'язки між екземплярами сутей. Діаграми ER-типів відображають зв'язки між типами сутей. Приклади діаграм ER-екземплярів і ER-типів наведені на рис.1 і рис.2, відповідно.

На діаграмах ER-типу (рис.2) суть відображається прямокутником, а зв'язок – ромбом. Нижче прямокутника розміщуються атрибут або набір атрибутів, що є ключем суті. Цифра (1 або n) вказує тип зв'язку. Місцезнаходження суцільного кола визначає чи обов'язково всі екземпляри даної суті повинні брати участь у зв'язках (коло знаходиться всередині прямокутника), чи не обов'язково (коло винесено за межі прямокутника). У

першому випадку говорять, що клас належності суті є обов'язковим, а в іншому - необов'язковим. У прикладах на рис. 1 і рис. 2 клас залежності суті 1 є необов'язковим, а клас належності суті 2 - обов'язковим. Тип зв'язку є 1:n, оскільки екземпляр 2 суті 1 зв'язаний з двома екземплярами суті 2.

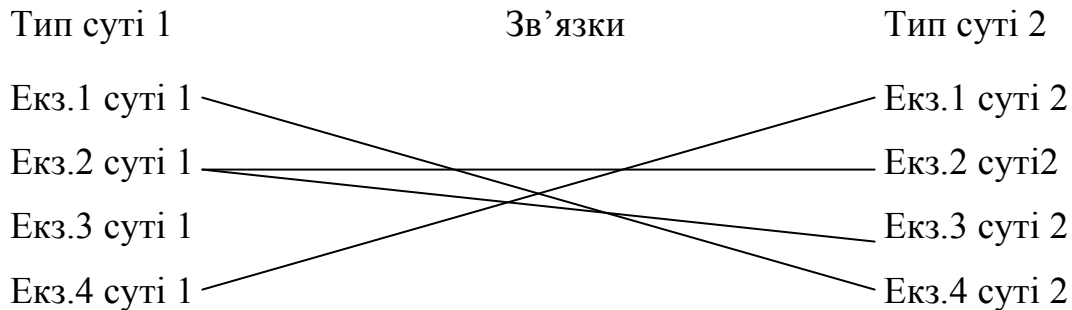


Рисунок 1 – Приклад діаграми ER-екземплярів

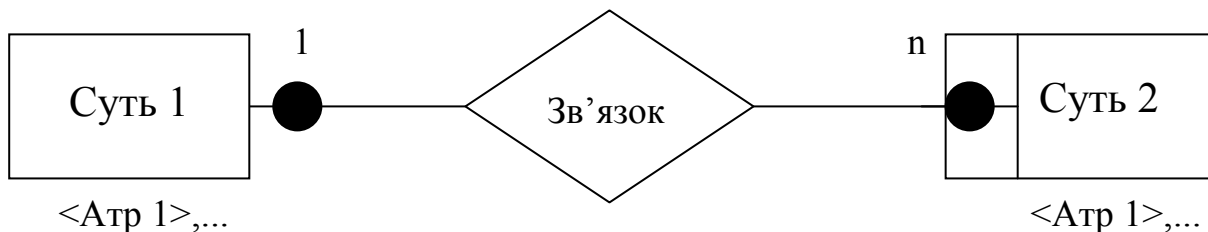


Рисунок 2 – Приклад діаграми ER-типів

Зазначимо, що хоча загальноприйнятого формату для графічного подання ER-діаграм немає, розбіжності у зображеннях, як правило, виявляються зовсім несуттєвими. Використаний нами стандартний спосіб подання запропонований Дж. Ульманом.

Першою задачею, яку необхідно розв'язати при розробці ER-моделі, є формування сутей, що необхідні для опису предметної області. Іншими словами, необхідно вказати ті типи об'єктів (тобто набори подібних об'єктів), про які в системі повинна накопичуватися інформація. Це означає, що за підсумками аналізу предметної області потрібно мати повну або достатньо повну уяву про запити, що реалізуються в системі. Разом з тим необхідно врахувати, що при кваліфікованій експлуатації системи у більшості випадків у користувача виникає бажання розширити систему запитів. У зв'язку з цим, при розробці ER-схеми, з одного боку, зручно твердо прив'язатися до обраної в результаті аналізу предметної області системи запитів, а з іншого боку, бажано розглядати задачу в більш широкому плані, з врахуванням перспектив подальшого нарощування можливостей системи.

Перечислимо суті, які використовуються в заданій предметній області, та визначимо ключі (атрибут чи групу атрибутів), які однозначно

ідентифікують поля даних сутей.

Особа(ідентифікаційний код).

Види кредитів(річні проценти).

Кредит(код кредиту).

Щомісячна оплата(дата, час).

Зв'язки можуть подаватися діаграмами ER-екземплярів і діаграмами ER-типів. Діаграми ER-екземплярів відображають зв'язки між екземплярами сутей. Діаграми ER-типів відображають зв'язки між типами сутей [1].

За допомогою діаграм ER-екземплярів визначимо зв'язки, які існують між вищенаведеними сутями. Визначимо типи зв'язків і характеристики кожного зв'язку.

Зв'язок **Особа – Кредит** поданий діаграмою екземплярів на рис. 3, з якого видно, що клас належності обох сутей є обов'язковим тому, що немає сенсу зберігати інформацію про особу, якщо вона не брала кредит, і не може бути такого, щоб кредит не мав особи, яка його має погасити. Тип зв'язку - 1:n, оскільки одна особа може взяти за своє життя не один кредит, проте її не нададуть наступний кредит, якщо вона не погасила попередній. І не може бути такого, щоб один і той самий кредит взяли декілька осіб.

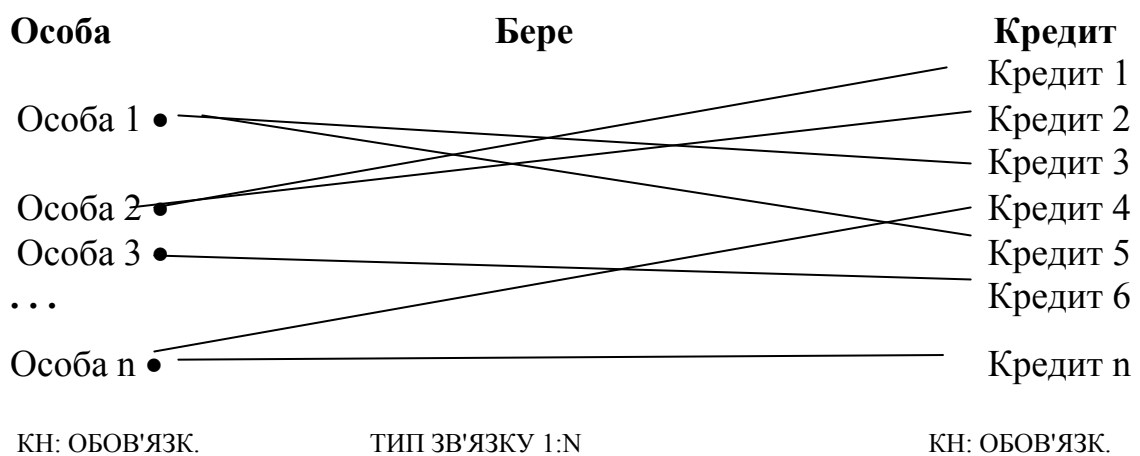


Рисунок 3- ER – Діаграма екземплярів сутей Особа і Кредит

Зв'язок **Види кредитів – Кредит** поданий діаграмою екземплярів на рис. 2, з якого видно, що клас належності суті Види кредитів є необов'язковим, тому що можливі такі види кредитів, які клієнти можуть і не взяти. Клас належності суті *Кредит* є обов'язковим, тому що кожен оформлений кредит має бути одним з видів суті види кредитів. Тип зв'язку - 1:n, оскільки один вид кредиту може оформити декілька осіб, проте один кредит може бути оформлений лише за одним із видів кредитів.

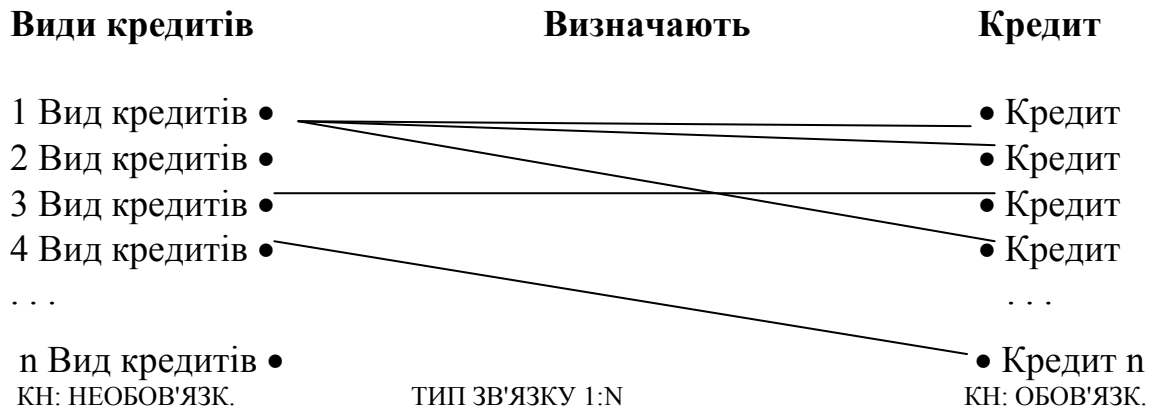


Рисунок 4 – ER-діаграма екземплярів сутей Види кредитів і Кредит

Зв'язок **Кредит – Щомісячна оплата** поданий діаграмою екземплярів на рис. 3, з якого видно, що клас належності суті Кредит є не обов'язковим, адже цілком можливо, що клієнт після отримання кредиту може його не погашати. Клас належності Щомісячна оплата є обов'язковим, адже якщо клієнт повертає деяку суму, то боргова сума повинна зменшитись на цю величину. Тип зв'язку - 1:n, адже кредит можна повертати малими сумами протягом вказаного періоду, але якщо клієнт повертає частину кредиту, то кредит на повернуту суму зменшується лише даному клієнтові.

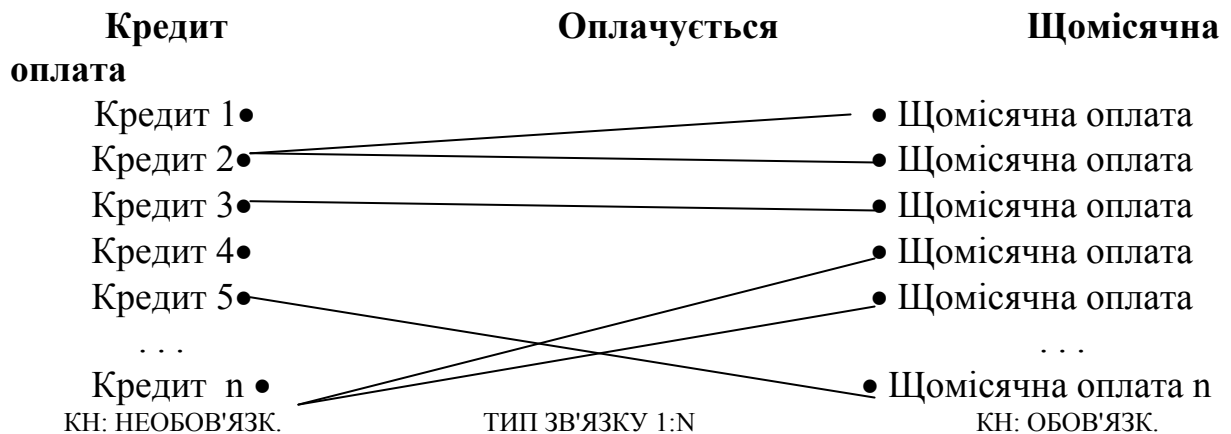


Рисунок 5 – ER-діаграма екземплярів сутей Кредит і Щомісячна оплата

Характеристики зв'язків і клас належності розглянутих сутей наведені в таблиці 3.

Таблиця 3 – Характеристики зв'язків предметної області “Контроль оплати по кредитах”

Ім'я суті 1	Ім'я суті 2	Тип зв'язку	Ім'я зв'язку	Клас належності
Особа	Кредит	1 : N	Бере	Обов., обов.
Види кредитів	Кредит	1 : N	Визначають	Необ., обов.
Кредит	Щомісячна оплата	1 : N	Оплачується	Необ., обов

З врахуванням вищезгаданого ER-модель досліджуваної предметної області приймає вигляд, що вказаний на рис. 6.

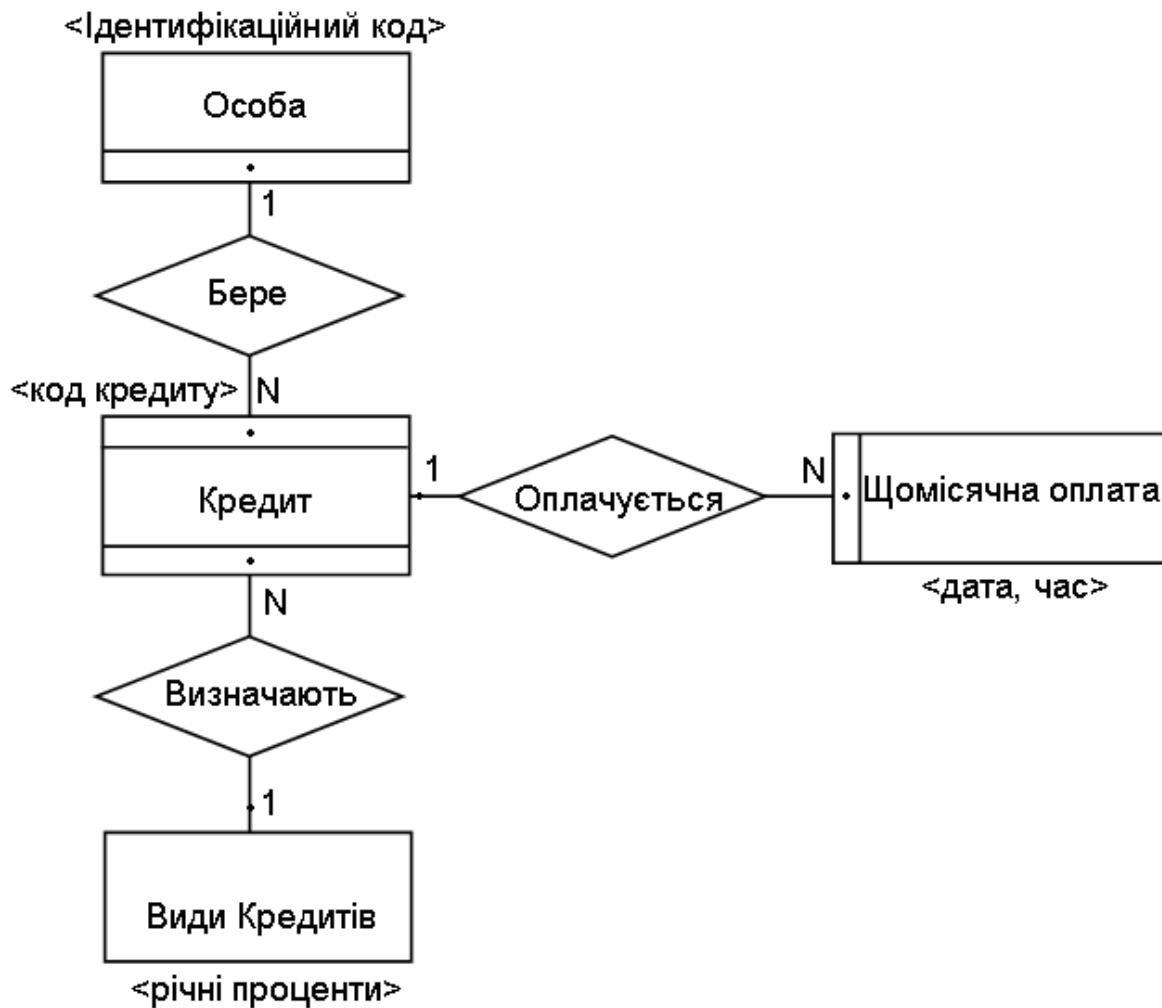


Рисунок 6 – Результуюча ER-модель предметної області “Контроль оплати по кредитах”

7.4 ПРОЕКТУВАННЯ НОРМАЛІЗОВАНИХ ВІДНОШЕНЬ

Під час розробки бази даних на основі універсальних відношень потрібно мати справу з великою надлишковістю даних. Крім того, може виникнути ряд проблем, що називаються аномаліями вставки, видалення та відновлення.

Визначимо різницю між дублюванням даних та надлишковим дублюванням даних. Для цього розглянемо приклад відношень наведений у таблиці 4.

Таблиця 4 – Приклад відношень з ненадлишковим дублюванням даних

Табельний номер робітника	Прізвище начальника
125	Бойко
138	Луцюк
195	Бойко
200	Луцюк

Прізвища начальників з'являються в таблиці 5 неодноразово (дублюються), проте зони не є надлишковими, оскільки видалення будь-якого прізвища з відношення приведе до загублення інформації.

Розглянемо тепер приклад відношення наведений в таблиці 5.

Таблиця 5 – Приклад відношень з надлишковим дублюванням даних

Табельний номер робітника	Прізвище робітника	Телефон начальника
125	Бойко	33333
138	Луцюк	22222
195	Бойко	33333
200	Луцюк	22222

У даному випадку ми маємо справу з надлишковим дублюванням даних. Причина надлишковості полягає у тому, що видаливши один з номерів телефону Бойко, ми можемо отримати інформацію про номер його телефону з іншого кортежа. Замінити в таблиці надлишкові телефони нулями не можна з таких причин:

- по-перше, при цьому знадобляться додаткові зусилля для визначення "реальних" значень нулів.
- по-друге, якщо у кортежі робітника з табельним номером 125

замінити значення телефону Бойко нулем, то при звільнені співробітника з табельним номером 200 виникне загублена інформація про номер телефону Бойко. Тому в таких випадках виконують заміну одного відношення двома. У нашому прикладі відношення таблиці 6 буде замінено відношенням таблиці 5 та таблиці 7.

Таблиця 6 – Приклад додаткового відношення, що використовується для видалення надлишкового дублювання даних.

Прізвище начальника	Телефон начальника
Бойко	33333
Луцюк	22222

Пояснимо тепер поняття аномалії вставки, видалення та відновлення. Аномалія вставка: полягає в тому, що якщо, наприклад, хоча б один з постачальників виробляє потрібну деталь, але не постачає її підприємству, у базу даних прийдеться включити кортеж з нульовим значенням атрибутів. Тобто, відсутня можливість включення в базу даних потенційних постачальників.

Аномалія відновлення обумовлена двома факторами:

- явною надлишковістю, яка полягає в тому, що, наприклад, інформація про постачальника, який брав участь в декількох поставках, буде з'являтися у базі даних декілька раз. Тобто, якщо у постачальника зміниться номер телефону, необхідно буде змінити його в декількох кортежах;

- неявною надлишковістю, яка полягає у тому, що, наприклад, зовсім різні постачальники можуть постачати одну і ту ж деталь, що має відповідно, один і той же код. Припустимо тепер, наприклад, що була проведена зміна коду деталі, про що один з постачальників повідомив підприємству, нічого не сказавши при цьому іншим постачальникам цієї ж деталі. Якщо тепер код деталі змінено, то зміняться функціональні залежності, що існують між атрибутами відношення.

Під функціональною залежністю атрибутів розуміють таке. Нехай дано атрибути А і В. Кажуть, що В функціонально залежить від А, якщо для кожного значення А в будь-який момент часу існує рівно одне пов'язане з ним значення В. Наприклад, в універсальному відношенні, що описує відеотеку, код користувача функціонально залежить від його паспортних даних і навпаки, паспортні дані функціонально залежать від коду користувача. Номер телефону функціонально залежить від коду користувача, але зворотної залежності не існує, якщо припустити, що користувачами відеотеки можуть бути декілька різних членів сім'ї, які

мешкають за однією і тією ж адресою.

Атрибути А і В можуть бути складеними, тобто являти собою не одиничні атрибути, а групи, які складаються з двох або більшої кількості атрибутів.

Найбільш розповсюдженою формою нотації функціональних залежностей є така:

$$A \Rightarrow B$$

Функціональні залежності визначають, виходячи з базових властивостей самих атрибутів. При цьому не можна обмежуватися звичайним переглядом атрибутів, що мають одне і те ж значення більш ніж в одному кортежі, оскільки функціональні залежності відображають семантичні (сміслові), а не синтактичні (структурні) зв'язки між атрибутами.

Виключення появи аномалій досягається шляхом видалення з відношення небажаних функціональних залежностей, в ході якого здійснюється декомпозиція відношень на ряд інших відношень, що є проєкціями вихідного. Таке видалення небажаних функціональних залежностей називається НОРМАЛІЗАЦІЄЮ - покроковим оборотним процесом заміни даного відношення (або даної сукупності відношень) іншою сукупністю відношень, в якій відношення мають більш просту і регулярну структуру. Апарат нормалізації відношень був розроблений Коддом, котрий виділив три нормальних форми подання відношень.

1. Введемо поняття простого і складного атрибута. Простим називається такий атрибут, значення котрого атомарні, тобто є нероздільними. У розглянутих нами прикладах універсальних відношень всі атрибути є простими. Складний атрибут може мати значення, що являє собою конкатенацію декількох значень одного або різних доменів.

Відношення називається зведеним до першої нормальної форми (1НФ), якщо всі його атрибути прості.

2. Введемо поняття повної функціональної залежності. Кажуть, що неключовий атрибут функціонально повно залежить від складеного ключа, якщо він функціонально залежить від ключа, але не знаходиться в функціональній залежності ні від будь-якої частини складеного ключа. замінять тільки в кортежах, що відносяться до даного постачальника, то правильний код деталі буде загублено, оскільки у відношенні будуть присутні два різних коди для позначення однієї і тієї ж деталі.

Відношення знаходиться в другій нормальної формі (2НФ), якщо воно знаходиться в 1НФ і кожний неключовий атрибут функціонально повно залежить від складеного ключа, а не від його частин. В нашій базі даних всі відношення крім Щомісячна оплата не містять складених ключів. Але у відношенні Щомісячна оплата (сума, дата, час) атрибут Сума функціонально повно залежить від складеного ключа. Тобто, можна

зробити висновок, що всі відношення даної бази даних знаходяться в 2НФ.

Відношення знаходиться в третій нормальній формі (3НФ), якщо воно знаходиться в 2НФ і кожен неключовий атрибут нетранзитивно залежить від початкового ключа. Визначимо поняття транзитивності функціональної залежності. Нехай X , Y та Z - три атрибути деякого відношення. При цьому має місце $X \rightarrow Y$ та $Y \rightarrow Z$, але обернена відповідність відсутня, тобто $Z \not\rightarrow Y$ або $Y \not\rightarrow X$. Тоді говорять, що Z транзитивно залежить від X . У відношеннях бази даних "Контроль оплати по кредитах" не існує транзитивних залежностей, тому можна зробити висновок, що всі відношення знаходяться в 3НФ.

В тих випадках, коли відношення має тільки один ключ і в ньому відсутні багатозначні залежності, 3НФ звільняє від надлишковості та аномалії виконання операцій включення, видалення та відновлення.

4. В тих випадках, коли у відношенні відсутні багатозначні залежності, але є два (або більше) можливих ключів, 3НФ може мати аномалії операцій.

Розглянемо відношення PROJECT ($\langle \text{Det}, \text{Proj} \rangle$, Post), що описує використання в проектах (Proj) деталей (Det), які поставляються постачальниками (Post). У проекті використовується декілька деталей, але кожна постачається тільки одним постачальником. Кожен постачальник обслуговує тільки один проект, але кожен проект може обслуговуватися декількома постачальниками. В цьому випадку мають місце такі функціональні залежності :

$\langle \text{Det}, \text{Proj} \rangle \Rightarrow \text{Post}$ та $\text{Post} \Rightarrow \text{Proj}$.

Таким чином, не дивлячись на те, що відношення PROJECT знаходиться в 3НФ, факт поставки постачальником деталі не можна завести в базу даних поки в проекті не почнуть їх використовувати. Якщо останній з типів деталей, що постачається для проекту постачальником, використаний, то будуть видалені і дані про постачальника. Якщо змінюється постачальник деякого типу деталей, необхідно виконувати повний перегляд цих деталей.

У таких випадках використовують "посилену" третю нормальну форму, котру називають також нормальною формою Бойса-Кодда (НФБК).

Атрибут (або комбінацію атрибутів), від якого будь-який інший атрибут залежить функціонально повно, називають детермінантним. Виходячи з того можна дати таке означення посиленої третьої нормальної форми: нормалізована схема відношення знаходиться в НФБК, якщо кожен детермінант є можливим ключем.

5. Існують також четверта та п'ята нормальні форми, обумовлені наявністю у відношенні багатозначних функціональних залежностей. Проте на практиці в більшості випадків обмежуються отриманням відношення в НФБК.

7.5 Отримання попередніх відношень за методом “Суть – зв’язок”

Загальний підхід до проектування баз даних на основі ER-методу включає в себе такі основні кроки:

- побудова діаграми ER-типу, що включає в свій склад всі суті і зв'язки даної предметної області;
- побудова набору попередніх відношень з вказанням передбачуваного початкового ключа для кожного відношення ;
- підготовка списку всіх атрибутів, що викликають інтерес (котрі не були перераховані як ключі сутей), і призначення кожного з цих атрибутів одному з попередніх відношень такі, щоб всі ці попередні відношення знаходились в НФБК. Для виконання цієї задачі необхідно визначити всі міжатрибутні функціональні залежності. У випадку, якщо не вдається привести відношення до НФБК або деяким атрибутам не вдається знайти логічно обґрунтованих місць, треба переглянути ER-діаграми на предмет видалення ускладнень, що виникли.

Попередні відношення формально генеруються з діаграми ER-типу на основі аналізу класу належності і ступені відношень сутей, на основі таких правил [14].

ПРАВИЛО 1. Якщо ступінь бінарних зв'язків 1:1 і клас належності обов'язковий для обох сутей, гарантується однократне появлення кожного значення сутей в будь-якому екземплярі відношень. Тобто, у відношенні ніколи не буде ні порожньої інформації, ні груп надлишкових даних, що повторюються.

Проте якщо клас належності однієї з сутей стане необов'язковим, то одного відношення буде недостатньо, оскільки у всіх кортежах, що містять інформацію про екземпляри однієї суті, які не мають зв'язків з екземплярами другої суті, з'являться пробіли.

Дійсно, єдине відношення містить кортежі, кожен з яких повинен включати атрибути, що описують обидві суті. Відсутність зв'язку між екземплярами сутей приведе у цьому випадку до того, що в кортежі буде відсутня інформація про атрибути, які описують одну з сутей.

ПРАВИЛО 2. Якщо ступінь бінарного зв'язку 1:1 і клас належності однієї суті є обов'язковим, а іншої - необов'язковим, то необхідна побудова двох відношень. При цьому ключ суті повинен служити первинним ключем для відповідного відношення. Крім того, ключ суті, для якої клас належності є обов'язковим, додається як атрибут у відношення, що виділене для суті з необов'язковим класом належності.

Формування двох відношень, кортежі кожного з яких включають лише опис атрибутів відповідної суті, приводить до виключення пробілів в відношеннях. При цьому зв'язок між сутями буде притримуватися завдяки включенню в одно з відношень ключових атрибутів другої суті.

ПРАВИЛО 3. Якщо ступінь бінарного зв'язку 1:1 і клас належності

обох сутей не є обов'язковим, то необхідно використовувати три відношення: по одному для кожної суті, ключі котрих служать первинні ключі відповідних відношень, і одне для зв'язку. Первинним ключем цього відношення може бути будь-яка з двох сутей. Серед своїх атрибутів відношення, що виділяється для зв'язку, повинно мати по одному ключу кожної суті.

Наявність в обох сутях "ізолюваних" екземплярів приводить до того, що ні в одному з відношень не буде гарантовано знаходитися значущі значення атрибута, що служать для зв'язку між відношеннями, тобто знову виникає проблема наявності у відношенні пробілів. У відношенні, що породжене зв'язком, пробіли завжди відсутні, оскільки увійшовши у нього кортежі не відображають "ізолювані" екземпляри сутей.

Під час роботи з відношеннями для бінарних зв'язків ступені 1:N фактором, що визначає вибір правила для генерації відповідного набору попередніх відношень, є клас належності n - зв'язаної суті. Клас належності 1-зв'язаної суті у цьому випадку на кінцевий результат не впливає.

ПРАВИЛО 4. Якщо ступінь бінарного зв'язку 1:N, і клас належності n -зв'язаної суті є обов'язковим, то досить використати по одному відношенню на кожну суть, при умові, що ключ кожної суті служать як первинний ключ для відповідного відношення. Додатково, ключ 1-зв'язаної суті повинен бути добавлений як атрибут у відношення, що відводиться n -зв'язній суті.

Очевидно, що використання двох відношень в цьому випадку дозволяє усунути дублювання інформації (багатократний опис атрибута 1-зв'язаної суті, зв'язаного з n атрибутами n -зв'язаної суті). Крім того, аналогічно правилу 2 забезпечується видалення з відношень пробілів.

ПРАВИЛО 5. Якщо ступінь бінарного зв'язку 1:N і клас належності n -зв'язаної суті є необов'язковими, то необхідно формування трьох відношень: по одному для кожної суті, причому ключ кожної суті служать як первинний ключ для відповідного відношення, і одного відношення для зв'язку. Зв'язок повинен мати серед своїх атрибутів ключ суті кожної з зв'язних сутей.

При ступені бінарного зв'язку $M:N$ без залежності від класу належності сутей завжди необхідно використовувати три відношення.

ПРАВИЛО 6. Якщо ступінь бінарного зв'язку $M:N$, то для зберігання даних потрібні три відношення: по одному для кожної суті, причому ключ кожної суті служать як первинний ключ для відповідного відношення, та одного відношення для зв'язку. Зв'язок повинен мати серед своїх атрибутів і ключ суті кожної зі зв'язних сутей. Виявлення в предметній області тристоронніх зв'язків приводить до необхідності використання чотирьох відношень.

ПРАВИЛО 7. У випадку наявності тристороннього зв'язку завжди використовуються чотири відношення: по одному для кожної суті.

Причому ключ кожної суті служить як первинний ключ для відповідного відношення, і одного відношення для зв'язку. Зв'язок повинен мати серед своїх атрибутів ключі суті кожної із зв'язних сутей.

Аналогічно для подання n-стороннього зв'язку необхідно використовувати n+1 попереднє відношення.

Використовуючи наведені правила, перерахунок атрибутів з універсального відношення та сформовану ER-модель предметної області, сформуємо попередні відношення, які описують задачу матеріального забезпечення заводу. Отримані попередні відношення зведемо в таблицю 7.

Таблиця 7 – Попередні відношення для опису предметної області “Контроль оплати по кредитах”

Ім'я зв'язку	Правило	Попередні відношення	Додаткові атрибути
Бере	4	R1(ідентифікаційний код, ...) *R2(код кредиту, ...)	ПІБ, адреса, рік народження, сума кредиту, дата надання, дата завершення, ідентифікаційний код
Визначають	4	R3(річні проценти, ...) *R4 (код кредиту, ...)	назва кредиту, максимальна сума сума кредиту, дата надання, дата завершення, ідентифікаційний код, річні проценти
Оплачується	4	R5(код кредиту, ...) R6(дата, час, ...)	сума кредиту, дата надання, дата завершення, ідентифікаційний код, річні проценти сума, код кредиту

У таблиці 7 наведено розподілення неключових атрибутів між

отриманими попередніми зв'язками. Знаком “*” помічені відношення, які потрібно видалити. Відношення R2 виключається з розглядання, оскільки вони повністю містяться у відношенні R5. Відношення R4 виключається з розглядання, оскільки воно повністю збігається з R5.

Після видалення відношень позначених * залишились відношення:

- R1(<ідентифікаційний код>, ПІБ, адреса, рік народження);
- R3(<річні проценти>, назва кредиту, максимальна сума);
- R5(<код кредиту>, сума кредиту, дата надання, дата завершення, ідентифікаційний код, річні проценти)
- R6(<дата, час>, сума, код кредиту).

Кінцеві відношення, що описують базу даних “Контроль оплати по кредитах”, подані таким набором:

Особа (ідентифікаційний код, ПІБ, адреса, рік народження);

Види кредитів (річні проценти, назва кредиту, максимальна сума) ;

Кредит (код кредиту, сума кредиту, дата надання, дата завершення, річні проценти, ідентифікаційний код) ;

Щомісячна оплата (дата, час, сума, код кредиту).

7.6 Нормалізація відношень методом декомпозиції

Одним з найбільш зручних методів проектування завеликих баз даних є метод декомпозиції [7], який включає в себе такі операції.

1. Розробка універсального відношення для бази даних.
2. Визначення всіх функціональних залежностей, які присутні між атрибутами універсального відношення.
3. Визначення того, чи знаходиться розглянуте відношення в НФБК. Якщо так, то проектування закінчується; якщо ні - відношення розбивається на два відношення.
4. Повторення кроків 2 та 3 для кожного нового відношення, отриманого в результаті декомпозиції.

Декомпозиція початкового відношення на два відношення здійснюється на основі аналізу наявності функціональних залежностей між атрибутами початкового відношення таким чином.

Нехай відношення R (A, B, C, D, E,...) не зведено до НФБК. Визначається функціональна залежність, наприклад, $C \rightarrow D$, з якої відомо, що вона є причиною того, що відношення не знаходиться в НФБК (C є детермінантом, але не є можливим ключем). Створюються два нових відношення: R1 (A, B, C, E, ...) та R2 (C, D), де залежна частина функціональної залежності опускається при формуванні R1 та повністю використовується під час формування відношення R2. Тепер необхідно перевірити, чи знаходяться в НФБК відношення R1 та R2. Про відношення R2 (C, D) говорять, що воно є проекцією відношення R. Цей тип декомпозиції називається декомпозицією без втрат при природному

з'єднанні. Даний прийом повторюється для кожного нового відношення, отриманого в результаті декомпозиції. Проектування закінчується, коли всі відношення будуть знаходитися в НФБК[1].

Розглянемо функціональні залежності, які присутні між атрибутами універсального відношення, що описує предметну область “Контроль оплати по кредитах” (рис.7).

< ідентифікаційний код > → адреса, ПІБ, рік народження;

< річні проценти > → назва кредиту, максимальна сума;

< код кредиту > → сума кредиту, дата надання, дата завершення, річні проценти, ідентифікаційний код;

< дата, час > → сума, код кредиту.

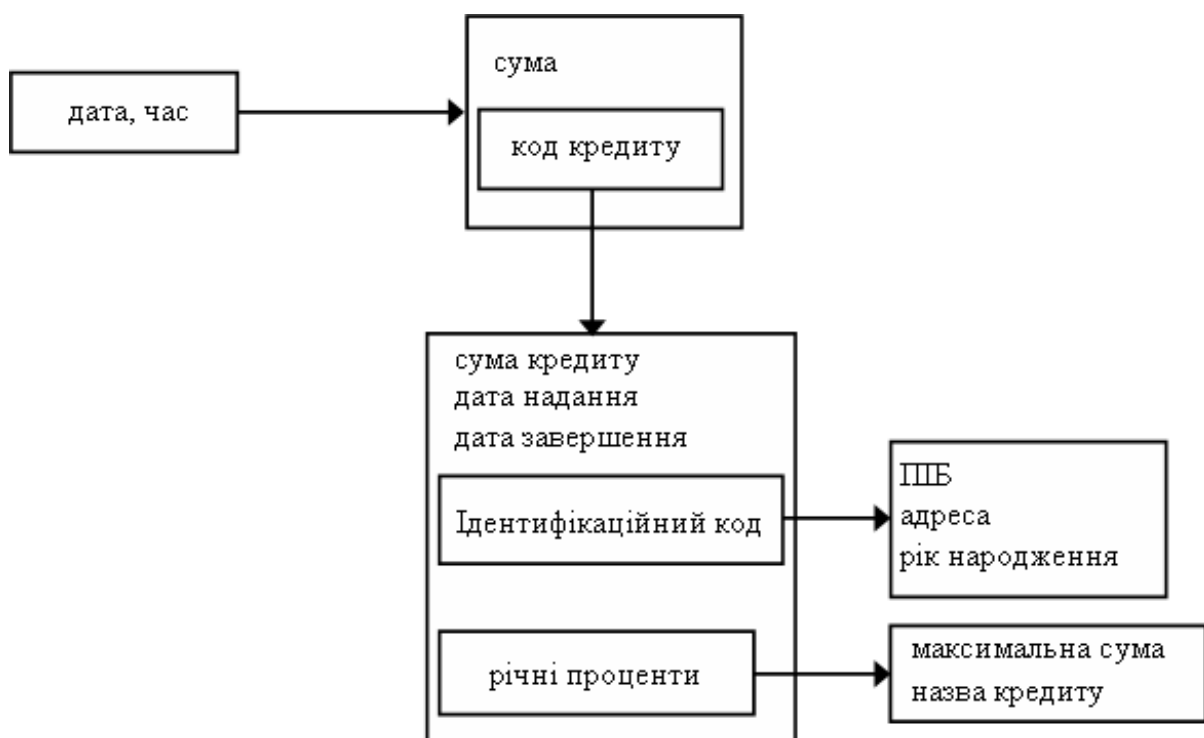


Рисунок 7 – Діаграма функціональної залежності

Універсальне відношення даної курсовій роботі має вигляд:

R (< ідентифікаційний код, річні проценти, код кредиту, дата, час >, адреса, ПІБ, рік народження, назва кредиту, максимальна сума, сума кредиту, дата надання, дата завершення, сума).

Детермінантами відношення є всі ліві частини функціональних залежностей < ідентифікаційний код >, < річні проценти >, < код кредиту >, < дата, час >.

На основі вищеперерахованих правил декомпозиції здійснимо послідовність операцій декомпозиції для отримання відношення в НФБК.

1. Виконаємо проєкціювання відношення R , яке породжується всіма функціональними залежностями, що містять атрибут ідентифікаційний код. Отримаємо відношення R1 і R2:

R1 (<річні проценти, код кредиту, дата, час>, ідентифікаційний код, назва кредиту, максимальна сума, сума кредиту, дата надання, дата завершення, сума);

R2 (<ідентифікаційний код>, адреса, ПІБ, рік народження).

Відношення R2 знаходиться в НФБК. А відношення R1 не знаходиться.

2. Виконаємо проєкціювання відношення R 1, яке породжується всіма функціональними залежностями, що містять атрибут код кредиту. У результаті отримаємо відношення R3, R4:

R3 (<річні проценти, дата, час>, код кредиту, назва кредиту, максимальна сума, сума);

R4 (<код кредиту>, сума кредиту, дата надання, дата завершення, ідентифікаційний код, річні проценти).

Відношення R4 знаходиться в НФБК. Для R3 ця умова не виконується.

3. Виконаємо декомпозицію R3 шляхом проєкціювання, породжуваного функціональною залежністю, для якої детермінантом є *дата, час* . В результаті отримаємо відношення R5, R6 :

R5 (<річні проценти >, назва кредиту, максимальна сума);

R6 (< дата, час >, сума, код кредиту).

Відношення R5, R6 знаходяться в НФБК, що свідчить про закінчення процедури декомпозиції.

Отже кінцеві відношення знайдені за допомогою методу декомпозиції, мають такий вигляд.

Особа (ідентифікаційний код , ПІБ, адреса, рік народження); (R2)

Види кредитів (річні проценти, назва кредиту, максимальна сума); (R5)

Кредит (код кредиту, сума кредиту, дата надання, дата завершення, річні проценти, ідентифікаційний код); (R4)

Щомісячна оплата (дата, час, сума, код кредиту). (R6)

7.7 Оцінка спроектованих НФБК-відношень

Перевірка НФБК-відношень, які розглядаються як кінцевий проєкт на предмет наявності невиявлених проблем, включає такі основні кроки :

1. Складаються списки функціональних залежностей дня кожного відношення. Ці списки перевіряються на двох напрямленнях:

- одна і таж функціональна залежність не повинна з'являтися більше, ніж в одному відношенні;

- набір функціональних залежностей, отриманих в результаті проєктування, повинен в точності збігатися з набором, присутнім в

мінімальному покритті, отриманому перед початком проектування. Іншими словами, потрібно буде довести можливість отримання підсумкового набору функціональних залежностей з мінімального покриття за допомогою правил виведення. Якщо хоча б одна з перевірок виявиться недостовірною (невірогідною), прийдеться аналізувати процес проектування для виявлення помилок і/або розглянути інші варіанти проектування.

2. Здійснюється перевірка на присутність надлишкових відношень. Відношення є надлишковим якщо :

- всі його атрибути можуть бути знайдені в одному або другому відношенні набору, що проектується;

- всі його атрибути можуть бути знайдені в відношенні, котре може бути отримане з інших відношень запропонованого проектного набору за допомогою серії JOIN - операцій над цими відношеннями. Якщо встановлена надлишковість відношення, його потрібно видалити з проектного набору. Наприклад, перший тип надлишковості присутній в такому наборі спроеКТованих відношень:

$R1(\langle A \rangle, B)$;

$R2(\langle B \rangle, C, Y, Z)$;

$R3(\langle A \rangle, B, K)$.

Очевидно, що відношення $R1$ є надлишковим, оскільки всі його атрибути присутні в відношенні $R3$.

Другий тип надлишковості можна проілюструвати на такому наборі відношень :

$R1(\langle A \rangle, C, X)$;

$R2(\langle D, K \rangle, F)$;

$R3(\langle D \rangle E, G, H)$;

$R4(\langle A, B \rangle, D)$;

$R5(\langle A, B, E \rangle, G)$;

$R6(\langle B \rangle, C, Y, Z)$.

Тут надлишковим є відношення $R7$, оскільки застосування операції JOIN над $R3$ та $R4$ (із загальним атрибутом D) дає відношення :

$R7(A, B, D, E, H)$,

котре містить всі атрибути, присутні в $R7$.

3. Розгляд відношень з практичної точки зору. Вивчається характер використання відношень в базі даних, що проектуються, і визначається чи будуть вони підтримувати ті типи операцій відновлення та запити, котрі передбачається використовувати.

В загальному випадку, в процесі проектування атрибути можуть згрупуватися нелогічно. Це вказує на можливу помилку, викликає в подальшому складності у реалізації запитів. Досвідчений проектувальник обов'язково слідкує за атрибутами в процесі проектування відношень і намагається осмислити, наскільки розумно вони групуються одне з одним.

Розглядаючи отримані при проектуванні бази даних матеріального забезпечення підприємства відношення, можна помітити, що:

- ні одна функціональна залежність (ФЗ) не повторюється більше одного разу;
- цей набір ФЗ є мінімальним.

Аналіз відношень показує, що не можна вказати серед них жодного, всі атрибути якого були б підмножиною атрибутів іншого відношення. Крім того, неможливо об'єднати будь-які два відношення так, щоб у результаті були отримані всі атрибути третього відношення. Тобто, жодне з відношень не є надлишковим, що свідчить про правильність проведення проектування.

Кінцеві відношення набудуть такого вигляду.

- Особа** (ідентифікаційний код, ПІБ, адреса, рік народження);
- Види кредитів** (річні проценти, назва кредиту, максимальна сума) ;
- Кредит** (код кредиту, сума кредиту, дата надання, дата завершення, річні проценти, ідентифікаційний код) ;
- Щомісячна оплата** (дата, час, сума, код кредиту).

8 РЕАЛІЗАЦІЯ ЗАПИТІВ І ВИХІДНИХ ФОРМ

8.1 Аналіз реалізованих БД запитів

Для збільшення функціональності та зручності у використанні бази даних «Контроль оплати по кредитах» було реалізовано такі запити.

- «СПИСОК БОРЖНИКІВ» За допомогою даного запиту можна отримати список боржників. До списку боржників відносяться ті клієнти, які не погасили кредит до зазначеного терміну.

- «СПИСОК НЕПОГАШЕНИХ КРЕДИТІВ» Розроблений запит дозволяє отримати список клієнтів, які ще не погасили кредит. Даний запит відрізняється від попереднього тим, що він, крім списку боржників, включає ще список клієнтів які не погасили кредит, проте кінцева дата погашення даних кредитів ще не настала.

- «СПИСОК КЛІЄНТІВ, ЯКІ ПОГАСИЛИ КРЕДИТ» Даний запит виводить список клієнтів які не мають кредитних зобов'язань. Тобто, клієнтам із даного списку банківська установа може знову надавати кредит. Даний запит виключає можливість надання клієнту кредит, якщо він не погасив попередній кредит, що є цілком очевидно.

- «СПИСОК КЛІЄНТІВ ЗА АДРЕСОЮ» Даний запит дозволяє визначити список клієнтів, які проживають за певною адресою. Він може бути корисний для визначення густини розподілу клієнтів по території.

8.2 Розробка вихідних форм

Форми відіграють важливе значення при розробці бази даних. Адже найчастіше саме за допомогою форм відбувається інтерактивний режим введення і виведення інформації в базі даних. Робота з формами є більш зручною і зменшує кількість помилок при роботі з базою даних. За допомогою форми можна об'єднати всю необхідну інформацію, позбавивши користувача бази даних пошуку її в різних місцях.

Розглянемо створені форми.

«ФОРМА РЕЄСТРАЦІЯ» - призначена для реєстрації нових клієнтів. На даній формі одночасно з реєстрацією відбувається оформлення першого кредиту, це зроблено для того, щоб в базі даних не було клієнтів, які не брали жодного кредиту.

«ФОРМА ОФОРМЛЕННЯ КРЕДИТУ» - служить для оформлення кредиту клієнтам, які вже отримували кредит. Також дана форма відслідковує боржників і не надає їм можливості знову отримувати кредит.

«ФОРМА ОСОБА» - за допомогою даної форми можна проглянути інформацію про кожного клієнта.

«ФОРМА ВИДИ КРЕДИТІВ»- дозволяє переглядати, редагувати та добавляти види кредитів, які клієнти можуть отримувати.

«ФОРМА ОПЛАТА КРЕДИТУ»- дозволяє клієнтові повертати кредит малими сумами. Саме за допомогою даної форми і здійснюється погашення кредиту. Форма надає детальну інформацію про клієнта і його кредит, а також автоматично вираховує непогашений залишок.

9 ЗРАЗОК ОФОРМЛЕННЯ ДОДАТКІВ

Додаток А



Рисунок А. 1 – Головна форма

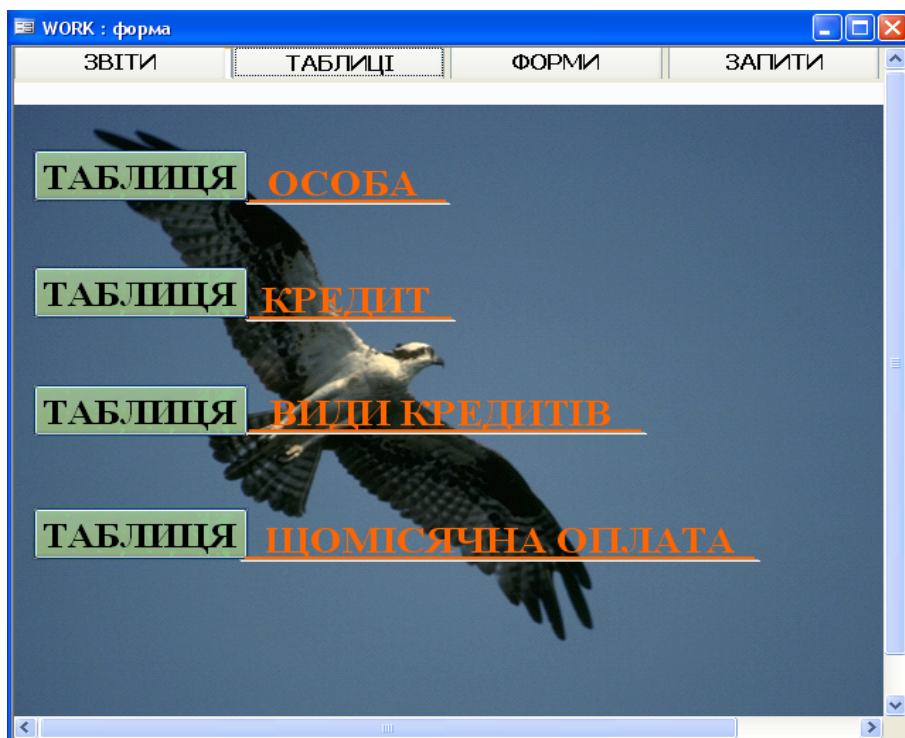


Рисунок А. 2 – Робоча форма із закладками

Види Кредитів

назва кредиту: Легкий

максимальна сума: 1 000,00 грн.

річні проценти: 4

Запись: 1 из 3

Рисунок А. 3 – Форма „ ВИДИ КРЕДИТИВ ”

Особа

ПІБ: Марчук О.Б.

адреса: М. Вінниця

дата народження: 01.01.1988

ідентифікаційний код: 323230

Запись: 1 из 13

Рисунок А. 4 – Форма „ ОСОБА ”

ОПЛАТА КРЕДИТУ

ПІБ: Марчук О.Б.

ідентифікаційний код: 323230

сума кредиту: 1 000,00 грн.

Непогашений залишок: 940 грн.

Погашена сума: 100,00 грн.

річні проценти: 4

	сума	дата	час
▶	100,00 грн.	19.05.2008	20:37:41
*	0,00 грн.	21.05.2008	20:15:59

Запись: 1 из 12

Рисунок А. 5 – Форма „ ОПЛАТА КРЕДИТУ ”

ФОРМА РЕЄСТРАЦІЯ

ідентифікаційний код

ПІБ

адреса

рік народження

Кредит

річні процент	сума кредиту	дата надан	дата заве
0	0,00 грн.	21.05.2008	

Запись: 1 из 1

Рисунок А. 6 – Форма „ РЕЄСТРАЦІЯ ”

ОФОРМЛЕННЯ КРЕДИТУ

ідентифікаційний код

ПІБ

Кредит

річні процент	сума кредиту	дата надан	дата заве
4	1 000,00 грн.	16.05.2006	03.03.2007
4	78,00 грн.	16.05.2008	01.01.2009
* 0	0,00 грн.	21.05.2008	

Запись: 1 из 2

Рисунок А. 7 – Форма „ ОФОРМЛЕННЯ КРЕДИТУ ”

Додаток Б

Особа : таблиця				
	ідентифікаційний код	ПІБ	адреса	рік народження
▶ +	323230	Марчук О.Б.	м. Вінниця	01.01.1988
+	123456782	Перун П.Н.	м. Вінниця	02.02.1980
+	123456789	Гавенко О.В.	м. Вінниця	07.09.1988
+	212123232	Горовенко О.П.	м. Вінниця	01.01.1988
+	232321237	Козочук А.Н.	м. Немирів	12.12.1950
+	343434330	Шевчук С.О.	м. Крижопіль	25.04.1956
+	435687989	Горобець Н.П.	м. Дніпропетров	01.02.1983
+	454545454	Кучієрук Д.М.	м. Вінниця	03.07.1976
+	544545455	Горошов Н.І.	м. Бар	01.06.1967
+	745454544	Пезенко М.П.	м. Одеса	01.04.1678
+	753434434	Ткачук Р.М.	м. Миколаїв	03.03.1989
+	765656556	Семененко О.І.	м. Донецьк	01.01.1976
+	928345678	Курчатов Н.О.	м. Житомер	01.01.1986
*	0			

Запись: 1 из 13

Таблиця Б. 1 – Особа

Кредит : таблиця						
	код кредиту	ідентифікаційний код	річні проценти	сума кредиту	дата надання	дата завершення
+	40	123456782	4	1 000,00 грн.	16.05.2006	03.03.2007
▶ +	41	123456782	4	78,00 грн.	16.05.2008	01.01.2009
+	42	212123232	4	44 444,00 грн.	10.05.2007	14.05.2007
+	53	753434434	4	44,00 грн.	17.05.2008	01.01.2001
+	55	765656556	4	400,00 грн.	17.05.2004	01.01.2005
+	56	745454544	4	1 000,00 грн.	17.05.2008	17.05.2009
+	57	212123232	4	44,00 грн.	17.05.2008	01.01.2002
+	60	323230	4	1 000,00 грн.	19.05.2008	19.05.2009
+	61	123456789	4	1 000,00 грн.	19.05.2008	19.05.2008
+	62	928345678	8	9 000,00 грн.	19.05.2008	10.05.2009
+	63	544545455	4	999,00 грн.	19.05.2008	19.05.2009
+	65	232321237	10	100 000,00 грн.	19.05.2008	19.05.2009
+	66	343434330	4	800,00 грн.	19.05.2008	19.05.2009
+	67	435687989	4	100,00 грн.	19.05.2008	19.05.2009
+	68	454545454	4	698,00 грн.	19.05.2008	19.05.2009
*	(Счетчик)	0	0	0,00 грн.	24.05.2008	

Запись: 2 из 15

Таблиця Б. 2 – Кредит

	назва кредиту	максимальна сума	річні проценти
▶ +	Легкий	1 000,00 грн.	4
+	Надійний	10 000,00 грн.	8
+	Макро	1 000 000,00 грн.	10
*		0,00 грн.	0

Запись: 1 из 3

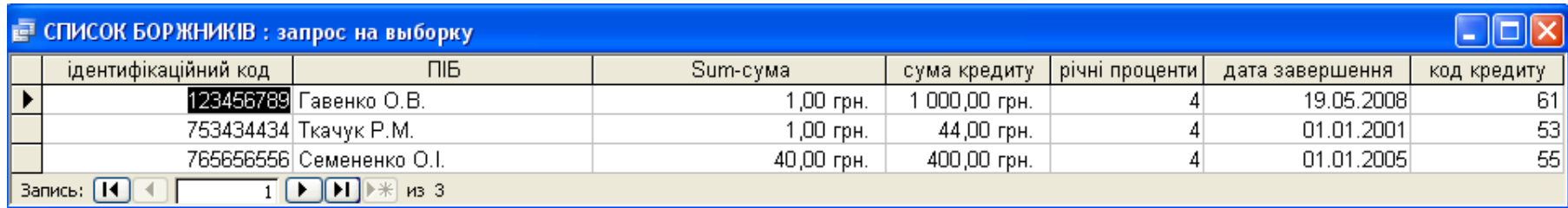
Таблиця Б. 3 – Види кредитів

	код кредиту	сума	дата	час
▶	42	34,00 грн.	16.05.2008	22:15:21
	42	4 444,00 грн.	17.05.2008	11:36:07
	42	11 742,00 грн.	17.05.2008	11:36:39
	42	54 545,00 грн.	17.05.2008	11:45:26
	40	100,00 грн.	17.05.2008	14:28:24
	55	40,00 грн.	17.05.2008	14:31:23
	56	5,00 грн.	17.05.2008	14:32:40
	40	434,00 грн.	17.05.2008	15:04:53
	60	100,00 грн.	19.05.2008	20:37:41
	40	434,00 грн.	19.05.2008	20:38:24
	40	200,00 грн.	19.05.2008	20:38:32
	61	1,00 грн.	19.05.2008	20:41:37
	62	100,00 грн.	19.05.2008	20:43:28
	63	100,00 грн.	19.05.2008	20:44:04
	41	45,00 грн.	19.05.2008	20:45:41
	41	34,00 грн.	19.05.2008	20:45:52
	65	1 000,00 грн.	19.05.2008	20:46:43
	66	100,00 грн.	19.05.2008	20:47:12
	67	3,00 грн.	19.05.2008	20:47:37
	68	56,00 грн.	19.05.2008	20:48:02
	53	1,00 грн.	19.05.2008	20:48:21
*	0	0,00 грн.	24.05.2008	11:59:28

Запись: 1 из 21

Таблиця Б. 4 – Щомісячна оплата

Додаток В

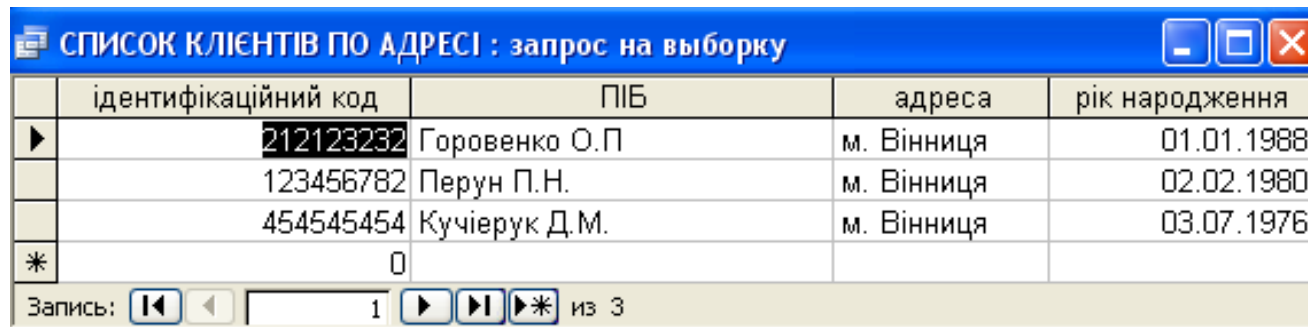


СПИСОК БОРЖНИКІВ : запит на вибірку

ідентифікаційний код	ПІБ	Sum-сума	сума кредиту	річні проценти	дата завершення	код кредиту
123456789	Гавенко О.В.	1,00 грн.	1 000,00 грн.	4	19.05.2008	61
753434434	Ткачук Р.М.	1,00 грн.	44,00 грн.	4	01.01.2001	53
765656556	Семененко О.І.	40,00 грн.	400,00 грн.	4	01.01.2005	55

Запись: 1 из 3

Рисунок В. 1 – Запит на вибірку „СПИСОК БОРЖНИКІВ”



СПИСОК КЛІЄНТІВ ПО АДРЕСІ : запит на вибірку

ідентифікаційний код	ПІБ	адреса	рік народження
212123232	Горовенко О.П	м. Вінниця	01.01.1988
123456782	Перун П.Н.	м. Вінниця	02.02.1980
454545454	Кучієрук Д.М.	м. Вінниця	03.07.1976
*	0		

Запись: 1 из 3

Рисунок В. 2 – Запит на вибірку „ СПИСОК КЛІЄНТІВ ЗА АДРЕСОЮ ”

СПИСОК КЛІЄНТІВ ЯКІ ПОГАСИЛИ КРЕДИТ : запит на вибірку						
ідентифікаційний код	ПІБ	Sum-сума	сума кредиту	річні проценти	дата завершення	
123456782	Перун П.Н.	1 168,00 грн.	1 000,00 грн.	4	03.03.2007	
212123232	Горovenko O.П	100 765,00 грн.	44 444,00 грн.	4	14.05.2007	

Запись: 1 из 2

Рисунок В. 3 – Запит на вибірку „ СПИСОК КЛІЄНТІВ, ЯКІ ПОГАСИЛИ КРЕДИТ”

СПИСОК НЕПОГАСЕНИХ КРЕДИТІВ : запит на вибірку						
ідентифікаційний код	ПІБ	Sum-сума	сума кредиту	річні проценти	код кредиту	
323230	Марчук О.Б.	100,00 грн.	1 000,00 грн.	4	60	
123456782	Перун П.Н.	79,00 грн.	78,00 грн.	4	41	
123456789	Гавенко О.В.	1,00 грн.	1 000,00 грн.	4	61	
232321237	Козочук А.Н.	1 000,00 грн.	300 000,00 грн.	10	65	
343434330	Шевчук С.О.	100,00 грн.	800,00 грн.	4	66	
435687989	Горобець Н.П.	3,00 грн.	100,00 грн.	4	67	
454545454	Кучіерук Д.М.	56,00 грн.	698,00 грн.	4	68	
544545455	Горошов Н.І.	100,00 грн.	999,00 грн.	4	63	
745454544	Пезенко М.П.	5,00 грн.	1 000,00 грн.	4	56	
753434434	Ткачук Р.М.	1,00 грн.	44,00 грн.	4	53	
765656556	Семененко О.І.	40,00 грн.	400,00 грн.	4	55	
928345678	Курчатov Н.О.	100,00 грн.	9 000,00 грн.	8	62	

Запись: 1 из 12

Рисунок В. 4 – Запит на вибірку „ СПИСОК НЕПОГАСЕНИХ КРЕДИТІВ”

Додаток Г

Види кредитів		
<i>назва кредиту</i>	<i>максимальна сума</i>	<i>річні проценти</i>
Легкий	1000,00 грн.	4
Макро	1000000,00 грн.	10
Надійний	10000,00 грн.	8

Рисунок Г. 1 – Звіт „ Види кредитів ”

Звіт про клієнтів			
<i>ідентифікаційний код</i>	<i>ПІБ</i>	<i>адреса</i>	<i>рік народження</i>
212123232	Горовенко Т. А.	м. Вінниця	01.01.1988
123456782	Перун П. Н.	м. Вінниця	02.02.1980
745454544	Пезенко П. Г.	м. Одеса	01.04.1678
765656556	Семененко О. І.	м. Донецьк	01.01.1976
753434434	Ткачук Р. М.	м. Миколаїв	03.03.1989
454545454	Кучіврук К. Л.	м. Вінниця	03.07.1976
435687989	Горобець А. А.	м. Дніпропетровськ	01.02.1983
343434330	Шевчук С. С.	м. Крижопіль	25.04.1956
232321237	Козочук А. К.	м. Немирів	12.12.1950
544545455	Горошов М. П.	м. Бар	01.06.1967
928345678	Курчаток П. Ю.	м. Житомир	01.01.1986
323230	Марчук О. Б.	м. Вінниця	01.01.1988
123456789	Гавенко О. В.	м. Вінниця	07.09.1988

Рисунок Г. 2 – Звіт про клієнтів

СПИСОК НЕПОГАШЕНИХ КРЕДИТІВ				
<i>ідентифікаційний код</i>	<i>ПІБ</i>	<i>погашена сума</i>	<i>сума кредиту</i>	<i>річні проценти</i>
323230	Марчук О. Б.	100,00 грн.	1000,00 грн.	4
123456782	Перун П. Н.	79,00 грн.	78,00 грн.	4
123456789	Гавенко О. В.	1,00 грн.	1000,00 грн.	4
232321237	Козочук А. К.	1000,00 грн.	1000000,00 грн.	10
343434330	Шевчук С. С.	100,00 грн.	800,00 грн.	4
435687989	Горобець А. А.	3,00 грн.	100,00 грн.	4
454545454	Кучіврук К. Л.	56,00 грн.	698,00 грн.	4
544545455	Горошов М. П.	100,00 грн.	999,00 грн.	4
745454544	Пезенко П. Г.	5,00 грн.	1000,00 грн.	4
753434434	Ткачук Р. М.	1,00 грн.	44,00 грн.	4
765656556	Семененко О. І.	40,00 грн.	400,00 грн.	4
928345678	Курчаток П. Ю.	100,00 грн.	9000,00 грн.	8

Рисунок Г. 3 – Звіт „ СПИСОК НЕПОГАШЕНИХ КРЕДИТІВ ”

СПИСОК БОРЖНИКІВ					
<i>ідентифікаційний код</i>	<i>ПІБ</i>	<i>заплатена сума</i>	<i>сума кредиту</i>	<i>річні проценти</i>	<i>дата завершення</i>
123456789	Гавенко О. В.	1,00 грн.	1000,00 грн.	4	19.05.2008
753434434	Ткачук Р. М.	1,00 грн.	44,00 грн.	4	01.01.2001
765656556	Семененко О. І.	40,00 грн.	400,00 грн.	4	01.01.2005

Рисунок Г. 4 – Звіт „ СПИСОК БОРЖНИКІВ ”

Кредитна діяльність				
<i>ідентифікаційний код</i>	<i>ПІБ</i>	<i>річні проценти</i>	<i>сума кредиту</i>	<i>дата завершення</i>
323230	Марчук О. Б.			
		4	1000,00 грн.	19.05.2009
123456782	Перун П. Н.			
		4	78,00 грн.	01.01.2009
		4	1000,00 грн.	03.03.2007
123456789	Гавенко О. В.			
		4	1000,00 грн.	19.05.2008
212123232	Горовенко Т. А.			
		4	44,00 грн.	01.01.2002
		4	44444,00 грн.	14.05.2007
232321237	Козочук А. К.			
		10	1000000,00 грн.	19.05.2009
343434330	Шевчук С. С.			
		4	800,00 грн.	19.05.2009
435687989	Горобець А. А.			
		4	100,00 грн.	19.05.2009
454545454	Кучіврук К. Л.			
		4	698,00 грн.	19.05.2009
544545455	Горошов М. П.			
		4	999,00 грн.	19.05.2009
745454544	Пезенко П. Г.			
		4	1000,00 грн.	19.05.2009
753434434	Ткачук Р. М.			
		4	44,00 грн.	01.01.2001
765656556	Семененко О. І.			
		4	400,00 грн.	01.01.2005
928345678	Курчаток П. Ю.			
		8	9000,00 грн.	10.05.2009

Рисунок Г. 5 – Звіт „ Кредитна діяльність ”

Додаток Д

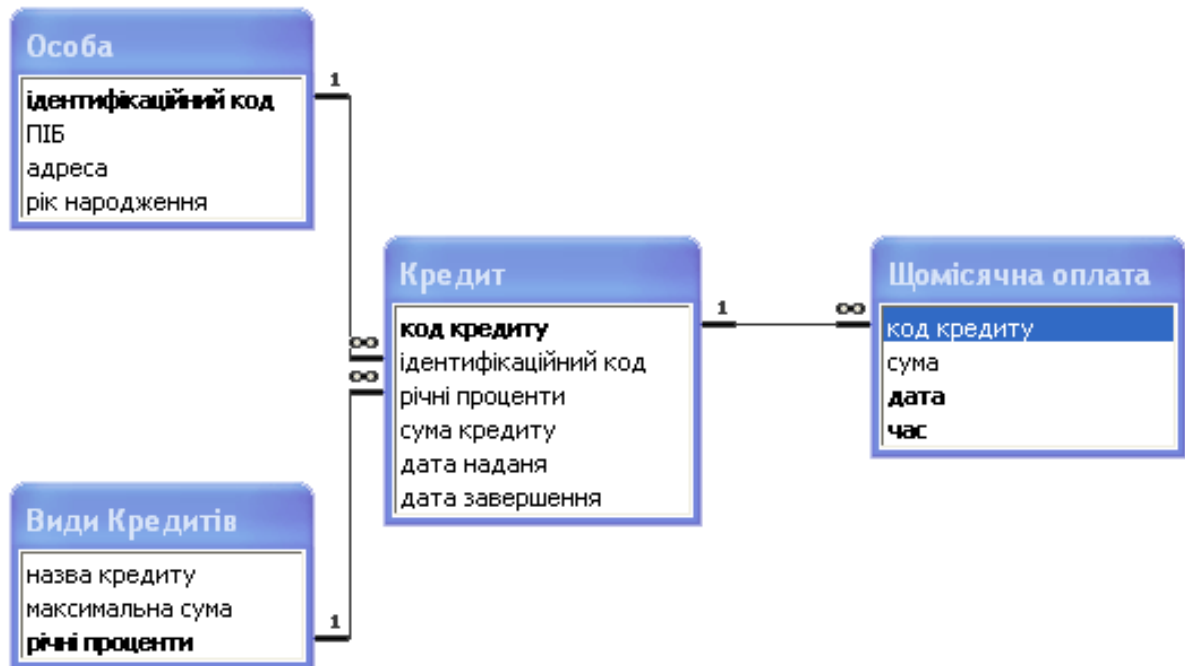


Рисунок Д. 1 – Схема даних

ЛІТЕРАТУРА

1. Базы и банки данных и знаний: Учебник для вузов/ Г. И. Ревунков, Э. Н. Самохвалов, В. В. Чистов; Под ред. В. Н. Четверякова.- М.: Высшая школа, 1992.
2. Берещанский Д. Г. Практическое программирование на dBase.- М.: Финансы и статистика, 1989.
3. Войтко В. В., Савинков В. П. Проектирование баз данных информационных систем.- М.: Финансы и статистика, 1989.
4. Гринберг Ф., Гринберг Р. Самоучитель программирования на входном языке СУБД dBase 3.- М.: Мир, 1989.
5. Гусева Т. И., Башин Ю. В. Проектирование баз данных в примерах и задачах.- М.: Радио и связь, 1992.
6. Дейт К. Введение в системы баз данных.- М.: Наука, 1980.
7. Джексон Г. Проектирование реляционных баз данных для использования с микро-ЭВМ: Пер. с англ./под ред. С. А. Платонова.- М.: Мир, 1991.
8. Дито М. М. Проектирование баз данных: Учебное пособие для вузов.- М.: Финансы и статистика, 1988.
9. Калинеченко Л. А. Методы и средства интеграции неоднородных баз данных.- М.: Наука, 1983.
10. Каррабрис Дж.-Д. Программирование в dBase 3 Plus: пер. с англ.- М.: Финансы и статистика, 1991.
11. Крамм Р. Системда управления базами данных dBase 2 и dBase 3 для персональных компьютеров.- М.: Финансы и статистика, 1986.
12. Куправа Т. М. Создание и программирование баз данных средствами СУБД dBase 3 Plus, Fox Base Plus, Clipper.- М.: Мир, 1991.
13. Мартэн Д. Базы данных: практические методы. Пер с англ.- М.: Финансы и статистика, 1979.
14. Мейер Д. Теория реляционных баз данных: Пер. с англ.- М.: Мир, 1987.
15. Нагао Маккао, Структуры и базы данных: Пер. с япон.- М.: Мир, 1991.
16. Озкарахан Э. Машины баз данных и управления ВД: Пер. с англ.- М.: Мир, 1989.
17. Потоцкий В. К. Работаем с системой Clipper.- М.: Мир, 1990.
18. Пунир О. Ф., Рамиро Перес В. Практические занятия по изучению

СУБД dBase 3+. Методические указания.- К.: КПИ, 1989.

19. Системы управления базами данных: Учебно-методическое пособие / Отв. ред. М. И. Лугачев.- М.: МГУ, 1991.

20. Справочное руководство по СУБД FoxBase+.- К.: МП Фирма "НТАС" 1991.

21. Теории Табл, Фрай Джейн. Проектирование структур баз данных: Пер. с англ./ В 2-х кн.- М.: Мир, 1985.

22. Ульман Дж. Основы систем баз данных: Пер. с англ.- М.: Финансы и статистика, 1983.

23. Хаббард Дж. Автоматизированное проектирование баз данных- Пер. с англ.- М.: Мир, 1984.

24. Цикризис Д., Лоховски Ф. Модели данных.- М.: Финансы и статистика, 1985.

Додаток А

Титульний аркуш

Міністерство освіти і науки України
Вінницький національний технічний університет
Інститут інформаційних технологій та комп'ютерної інженерії

Кафедра ПЗ

БАЗА ДАНИХ «_____»

Пояснювальна записка
з дисципліни «Основи баз даних і знань»
до курсової роботи за спеціальністю
«Програмне забезпечення автоматизованих систем»
08-26.ОБДіЗ.077.00.000 ПЗ

Керівник курсової роботи

ст. викл. _____

(прізвище та ініціали)

(підпис)

«___» _____ 2010р.

Розробив студент гр. _____

(підпис, прізвище та ініціали)

«___» _____ 2010р.

Вінниця ВНТУ 2010

Додаток Б

Сторінка індивідуального завдання

Міністерство освіти і науки України
Вінницький національний технічний університет
Інститут інформаційних технологій та комп'ютерної інженерії

ЗАТВЕРДЖУЮ
Зав. кафедри ПЗ., д.т.н.
А. М. Петух
«___» _____ 2010р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу з дисципліни «Основи баз даних та знань»

Студента _____, факультету КІ, групи _____

ТЕМА: «Розробка бази даних в середовищі MS Access»

Розробити базу даних заданої предметної області в середовищі Access 2000/XP.

Вимоги до бази даних

- ступінь універсального відношення, не менше 12;
- потужність універсального відношення, не менше 12;
- кількість «сутей» ER-діаграми, не менше 4;
- кількість попередніх відношень, не менше 4;
- форма нормалізації первинних відношень, не менше 3;
- кількість таблиць БД, не менше 4;
- кількість вихідних норм, не менше 4;
- кількість запитів, що реалізуються, не менше 4;
- робота користувача з БД реалізується тільки за допомогою форм;
- обов'язковий захист БД від несанкціонованого доступу;
- вхідне середовище створення MS Access;

База даних повинна забезпечувати

- роботу користувача у режимі кольорового меню;
- введення, видалення, оновлення інформації;

- видачу відповідей на запити щодо вибору користувача на екран дисплея, або ж пристрій друку;
- режим підказка для роботи користувача;
- збереження записів;
- контроль пріоритетів доступу до окремих блоків даних шляхом накладання пароля.

Пояснювальна записка формується у вигляді документа Word 2000/XP за таких вимог:

- аркуш формату А4;
- поля: 2 см – зверху і знизу, 2,5 – зліва, 1 см – справа;
- шрифт – Times New Roman 14;
- інтервал між рядками – 1,5;
- відступ першого рядка – 1,25;
- обов'язкова наявність заголовків розділів і підрозділів.

Структура пояснювальної записки

- титульний лист;
- зміст;
- індивідуальне завдання;
- вступ;
- аналіз предметної області;
- розробка структури бази даних;
- розробка форм;
- розробка запитів;
- розробка звітів;
- висновки;
- література.

Предметна область:
« _____ ».

Проектування бази даних

Дата видачі «__» _____ 2010р.

Керівник: _____

Завдання отримав: _____

Додаток В

Оформлення тексту пояснювальної записки

Нумерація сторінок курсової роботи наскрізна – титульний листок – перша сторінка і т.д. Звичайно на титульному листку, на листку індивідуального завдання, анотації номер сторінки не ставиться. Номер ставиться в правому верхньому кутку листка.

Зміст не містить інформації про елементи записки, які знаходяться в змісті. Тобто перший пункт, який знаходиться у змісті – це вступ.

Об'єм пояснювальної записки 25-30 сторінок (без додатків). Відступи від країв аркуша: зверху, знизу і зліва – 20 мм; справа – 10 мм. Абзац – 5 знаків;

Заголовки структурних частин, розділів виконують великими літерами посередині рядка, всі інші з абзацу малими літерами починаючи з великої.

Шрифт – 14 пт . Міжрядковий інтервал – 1.5 пт.

Стосовно ілюстрацій: на всі ілюстрації в тексті ПЗ мають бути посилання. Посилання виконують за формою: “...показано на *рисунок 3.1.*” або в дужках за текстом (*рисунок 3.1*), на частину ілюстрації: “... показані на *рисунок 3.2, б*”. Посилання на раніше наведені ілюстрації дають зі скороченим словом “*дивись*” відповідно в дужках (*див. рисунок 1.3*). Всі ілюстрації в ПЗ називають *рисунками* і позначають під ілюстрацією симетрично до неї за такою формою: “*Рисунок 3.5 – Найменування рисунка*”. Крапку в кінці не ставлять, знак переносу не використовують. Якщо найменування *рисунка* довге, то його продовжують у наступному рядку, починаючи від найменування.

Нумерують ілюстрації в межах розділів, вказуючи номер розділу і порядковий номер ілюстрації в розділі, розділяючи крапкою.

Якщо частини ілюстрації не вміщуються на одній сторінці, то їх переносять на наступні сторінки. В цьому випадку під початком ілюстрації вказують повне її позначення, а під її продовженнями позначають “*Рисунок 3.2 (продовження).*”

Таблицю розміщують симетрично до тексту після першого посилання на даній сторінці або на наступній, якщо на даній вона не уміщується і таким чином, щоб зручно було її розглядати без повороту або з поворотом на кут 90° за годинниковою стрілкою.

ГОСТ 2.105-95 та ДСТУ 3008-95 пропонують такий запис таблиці:

Таблиця _____ - _____
(номер) (назва таблиці)

На всі таблиці мають бути посилання за формою: “*наведено в таблиці 3.1*”; “ ... в *таблицях 3.1 – 3.5*” або в дужках по тексту (*таблиця 3.6*). Посилання на раніше наведену таблицю дають з скороченим словом

”дивись” (див. таблицю 2.4) за ходом чи в кінці речення. Графу “№ п/п” в таблицю не включають.

Найменування (при його наявності) таблиці розміщують тільки над першою частиною, а над іншими частинами зліва пишуть “Продовження таблиці 4.2” без крапки в кінці. Продовження таблиці повинно починатись з шапки таблиці – тобто рядка з найменуванням стовпців.

Перелік посилань повинен містити біля 10 посилань. Список містить перелік літературних джерел, на які повинні бути обов’язкові посилання в тексті пояснювальної записки. Література (книги, статті, патенти, журнали) в загальний список записується в порядку посилання на неї в тексті. Посилання на літературу наводять в квадратних дужках [...], вказуючи порядковий номер за списком. Літературу записують мовою оригіналу. В списку кожну літературу записують з абзацу, нумерують арабськими цифрами, починаючи з одиниці.

Висновки є заключною частиною, підсумком прийнятого конструкторського рішення виконаного проекту із зазначенням досягнутих параметрів та переваг об’єкта в порівнянні з існуючими аналогами, з можливими рекомендаціями прикладного застосування та шляхами (перспективами) удосконалення спроектованого об’єкта. Обов’язково слід зазначити, чого ви навчились, розробляючи курсову.

Додатки оформлюють як продовження документа на його наступних сторінках, розташовуючи в порядку посилань на них у тексті ПЗ. До додатків також відносять ілюстрації, таблиці, тексти допоміжного характеру. Посилання на додатки в тексті ПЗ дають за формою:

“... наведено в додатку А”, „... наведено в таблиці В.5” або (додаток Б); (додатки К, Л).

Кожен додаток необхідно починати з нової сторінки, вказуючи зверху посередині рядка слово “Додаток” і через пропуск його позначення. Додатки позначають послідовно великими українськими буквами, за винятком букв Г, Є, З, І, Ї, Й, О, Ч, Ь, наприклад, Додаток А, Додаток Б і т. д. Якщо додатків більше ніж букв, то продовжують позначати арабськими цифрами. Дозволяється позначати додатки латинськими буквами, за винятком букв І і О. Під позначенням для обов’язкового додатку пишуть в дужках слово (обов’язковий), а для інформативного – (довідковий).

Кожен додаток повинен мати тематичний (змістовний) заголовок, який записують посередині рядка малими літерами, починаючи з великої. При наявності основного напису – заголовок записують у відповідній графі.

Ілюстрації, таблиці, формули нумерують в межах кожного додатка, вказуючи його позначення: “Рисунок Б.3 - Найменування”; “Таблиця В.5 – Найменування” і т. п.

Навчальне видання

Методичні вказівки
до виконання курсової роботи з дисципліни
"Організація баз даних і знань"
для студентів напряму підготовки 6.050103
"Програмна інженерія"
денної та заочної форм навчання

Редактор В. Дружиніна
Коректор З. Поліщук

Укладачі: Олександр Никифорович Романюк
Алла Василівна Денисюк

Оригінал-макет підготовлено А. В. Денисюк

Підписано до друку
Формат 29,7 × 42¹/₄. Папір офсетний.

Гарнітура Times New Roman.
Друк різнографічний. Ум. друк. арк.
Наклад прим. Зам. №

Вінницький національний технічний університет,
науково-методичний відділ ВНТУ.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.

Свідоцтво суб'єкта видавничої справи
Серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-81-59.

Свідоцтво суб'єкта видавничої справи
Серія ДК № 3516 від 01.07.2009 р.