

Міністерство освіти і науки України
Донецький національний університет імені Василя Стуса
Факультет математики та інформаційних технологій

П. В. Римар, К. В. Смоктьй, О. Д. Смоктьй

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ
ЩОДО ВИКОНАННЯ ЛАБОРАТОРНИХ ЗАВДАНЬ
З КУРСУ «БАЗИ ДАНИХ ТА ІНФОРМАЦІЙНІ СИСТЕМИ»
(ПРОЕКТУВАННЯ БАЗ ДАНИХ, РЕЛЯЦІЙНА АЛГЕБРА)**

Вінниця 2019

УДК 004.652.43(072)

Р 51

*Рекомендовано до друку вченою радою
факультету математики та інформаційних технологій
(протокол № 10 від 20.06.2019 р.)*

Автори: *П. В. Римар*, старший викладач кафедри прикладної математики і теорії систем управління;
К. В. Смоктій, доцент кафедри прикладної математики і теорії систем управління;
О. Д. Смоктій, доцент кафедри прикладної механіки і комп'ютерних технологій.

Рецензенти: *О. М. Анісімова*, д-р екон. наук, професор, завідувач кафедри інформаційних систем управління;

Ю. С. Антонов, канд. фіз.-мат. наук, доцент, доцент кафедри прикладної математики і теорії систем управління.

Римар П. В., Смоктій К. В., Смоктій О. Д.

Р 51 Методичні рекомендації щодо виконання лабораторних завдань з курсу «Бази даних та інформаційні системи» (проекування баз даних, реляційна алгебра) / П. В. Римар, К. В. Смоктій, О. Д. Смоктій. Вінниця: ДонНУ імені Василя Стуса, 2019. 40 с.

У методичних рекомендаціях розглянуті принципи проектування систем баз даних та операції реляційної алгебри. Розв'язані типові приклади з тем та наведені задачі для засвоєння матеріалу.

Методичні рекомендації призначені для студентів галузей знань 11 «Математика і статистика» та 12 «Інформаційні технології».

УДК 004.652.43(072)

© Римар П. В., 2019
© Смоктій К. В., 2019
© Смоктій О. Д., 2019
© ДонНУ імені Василя Стуса, 2019

ЗМІСТ

ВСТУП	4
РОЗДІЛ 1. ПРОЕКТУВАННЯ СТРУКТУРИ БАЗИ ДАНИХ.....	5
1.1 Теоретичні відомості	5
1.2 Задачі для самостійного розв'язання	11
РОЗДІЛ 2. ОПЕРАЦІЇ РЕЛЯЦІЙНОЇ АЛГЕБРИ.....	23
2.1 Теоретичні відомості	23
2.2 Задачі для самостійного розв'язання	30
РЕКОМЕНДОВАНА ЛІТЕРАТУРА	40

ВСТУП

Курс «Бази даних та інформаційні системи» є одним з найважливіших в підготовці фахівців галузей знань 11 «Математика і статистика» та 12 «Інформаційні технології». При постійному зростанні обсягу і вимог до оперативності обробки інформації рішення різноманітних проблем управління, обліку, аналізу та контролю досягається застосуванням сучасних комп'ютерних технологій. Отже, проблема ефективного представлення інформації в пам'яті комп'ютера безумовно актуальна. Пропоновані рекомендації містять приклади типових задач для проектування баз даних та операцій реляційної алгебри.

Основну увагу в методичних рекомендаціях приділено якісному та ефективному проектуванню структури бази даних та операціям реляційної алгебри.

РОЗДІЛ 1. ПРОЕКТУВАННЯ СТРУКТУРИ БАЗИ ДАНИХ

1.1. Теоретичні відомості

Визначення 1.1. *Інформаційна система* – це програмний комплекс, що забезпечує виконання таких функцій: підтримка надійного зберігання інформації у пам'яті комп'ютера, виконання специфічних для цього додатка перетворень інформації та / або обчислень, надання користувачам зручного інтерфейсу, який легко опанувати.

Інформаційні системи можна розробляти у два способи:

1. Засобами звичайних мов програмування, зі збереженням інформації у файлах даних. Такі системи називаються *файловими*.

2. Використання концепції баз даних. Такі системи називаються *системами баз даних*.

Визначення 1.2. *Система баз даних* – це комп'ютеризована система збереження записів, основна мета якої зберігати інформацію та надавати її на вимогу. До її складу включають дані (*база даних*), апаратне забезпечення (сервер, робочі станції), програмне забезпечення (СУБД, прикладні програми), користувачі (вказані тільки кінцеві користувачі).

Визначення 1.3. *База даних* – це сукупність екземплярів різних типів записів і відношень між записами, агрегатами даних, елементами даних.

Визначення 1.4. *База даних* – сукупність даних, організованих відповідно до концепції, яка описує характеристику цих даних і взаємозв'язки між її елементами. В загальному випадку база даних містить схеми, таблиці, подання, збережені процедури та інші об'єкти.

Визначення 1.5. *Система управління базами даних (СУБД)* – це програмне забезпечення, що надає можливість створювати, надійно зберігати, оновлювати та шукати відповідну інформацію в базах даних з контролем доступу до даних.

Визначення 1.6. *Однокористувацька система або система з одним користувачем* – це система, в якій в один і той же час до бази даних може отримати доступ не більше одного користувача.

Визначення 1.7. *Багатокористувацька система або система з декількома користувачами* – система, в якій до бази даних одночасно можуть отримати доступ відразу декілька користувачів.

Визначення 1.8. *Інфологічна модель даних (семантична модель)* – це опис вмісту бази даних, виконаний з використанням природньої мови, математичних формул, таблиць, графіків та інших засобів, зрозумілих людям, що працюють над проектуванням бази даних.

Визначення 1.9. *Схема баз даних* – це структура системи баз даних, описана формальною мовою, яка підтримується системою управління базами даних і належить до організації даних з розподілом на таблиці.

Визначення 1.10. *Поле (атрибут, елемент даних, властивість)* – найменша (неподільна) одиниця іменованих даних.

Визначення 1.11. *Домен* – це іменована множина скалярних значень одного і того ж типу, з якої беруться значення для визначених полів (атрибутів) відношення.

Визначення 1.12. *Кортежем* називається група взаємопов'язаних елементів даних, що відповідає рядку таблиці.

Визначення 1.13. *Відношення R* , визначене на множині доменів D_1, D_2, \dots, D_n (не обов'язково різних), містить дві частини: заголовок і тіло. **Заголовок** містить фіксовану множину атрибутів або, точніше, пар <ім'я атрибута, ім'я домену>: ($\langle A_1:D_1 \rangle, \langle A_2:D_2 \rangle, \dots, \langle A_n:D_n \rangle$), причому кожен атрибут A_j відповідає одному і тільки одному з доменів D_j ($j=1,2,\dots,n$), що лежить в його основі. Всі імена атрибутів A_1, A_2, \dots, A_n – різні. **Тіло** містить множину кортежів. Кожен кортеж, в свою чергу, містить множину пар <ім'я атрибута: значення атрибута>: $\{ \langle \langle A_1:v_1 \rangle, \langle A_2:v_2 \rangle, \dots, \langle A_n:v_n \rangle \rangle, i=1,2,\dots, m$, де m – кількість кортежів в множині. У кожному кортежі є одна така пара <ім'я атрибута: значення атрибута>, тобто $\langle A_j:v_{ij} \rangle$, для кожного атрибута

A_j в заголовку. Для будь-якої такої пари $\langle A_j: v_{ij} \rangle$ v_{ij} є значенням з унікального домену D_j , який пов'язаний з атрибутом A_j .

Визначення 1.14. *Потенційним ключем* K для відношення R назвемо підмножину атрибутів R , що має такі властивості: унікальності (тобто немає двох різних кортежів у відношенні R з однаковим значенням K) та ненадлишковості (тобто жодне з підмножин K не наділене властивістю унікальності).

Визначення 1.15. *Первинний ключ* – це унікальний ідентифікатор для таблиці, тобто стовпець (або їхня комбінація), для якого в будь-який момент часу не існує двох рядків, які містять однакове значення в цьому стовпці (їхній комбінації).

Визначення 1.16. *Первинний ключ* – це один з потенційних ключів, який обирається в ролі первинного ключа в базовому відношенні.

Визначення 1.17. *Альтернативними ключами* називаються всі потенційні ключі, які залишилися після обрання первинного.

Кожне відношення обов'язково повинно мати первинний ключ для однозначної ідентифікації кожного рядку. Назва поля для збереження значення первинного ключа повинна мати в своїй назві ID та назву відповідного відношення (таблиці), що надасть можливість його однозначно інтерпретувати.

Визначення 1.18. Зовнішнім ключем FK в базовому відношенні R_2 назвемо підмножину атрибутів R_2 , для якої виконуються дві властивості:

1. Існує базове відношення R_1 (R_1 і R_2 не обов'язково різні) з потенційним ключем СК.

2. Кожне значення FK в поточному значенні R_2 або є null-значенням, або збігається зі значенням СК деякого кортежу в поточному значенні R_1 .

Бази даних містять таблиці, які пов'язані між собою різними зв'язками. Зв'язок представляє собою асоціацію сутностей різних типів. При виділенні зв'язку виокремлюють головну (батьківську) таблицю та залежну (дочірню) таблицю. Дочірня таблиця залежить від батьківської.

Для організації зв'язку використовують зовнішні ключі. Зовнішній ключ – це один або декілька стовпців з однієї таблиці, який одночасно є потенційним ключем іншої таблиці. Зовнішній ключ необов'язково повинен відповідати первинному ключу з головної таблиці. Як правило, зовнішній ключ із залежної таблиці вказує на первинний ключ з головної таблиці.

Зв'язок між таблицями може бути:

Один–до–одного. Цей тип зв'язку зустрічається рідко. В цьому випадку об'єкту однієї сутності можна віднести тільки один об'єкт іншої сутності. *Наприклад:* на певних сайтах користувач може мати тільки один блог. Тоді виникає відношення один користувач–один блог.

Один–до–багатьох. Це тип зустрічається найчастіше. В ньому декілька рядків з дочірньої таблиці відповідають одному рядку в батьківській таблиці. *Наприклад:* у однієї людини можуть бути декілька паспортів. В цьому випадку таблиця з інформацією про людей є батьківською, а таблиця паспортів – дочірньою.

Багато–до–багатьох. При цьому типі зв'язку один рядок з таблиці А може бути пов'язаний з множиною рядків таблиці Б. В свою чергу, один рядок таблиці Б може бути пов'язаний з множиною рядків таблиці А. *Наприклад:* один студент може відвідувати декілька курсів, відповідно на один курс можуть записатись декілька студентів. В реляційних базах даних відсутній механізм реалізації зв'язку багато–до–багатьох між двома таблицями. Це робиться з використанням (додаванням нової) проміжної таблиці, яка містить первинні ключі як з таблиці А, так і таблиці Б. Іноді дані з проміжної таблиці є окремими сутностями з відповідними властивостями. Після введення проміжної таблиці отримаємо два зв'язки один-до-багатьох.

Проектування структури бази даних – один з важливих етапів розробки інформаційної системи. База даних містить в собі таблиці, поля, відношення, ключі та інше. Розглянемо процес проектування схеми на конкретному прикладі.

Для визначеності будемо припускати такі обмеження на імена об'єктів, що використовуються в базі даних:

- 1) домени мають унікальні імена в базі даних;
- 2) імена відношень (таблиць) унікальні в базі даних;
- 3) імена атрибутів (поля) унікальні у відношенні, що їх містить.

Визначення 1.19. Основними цілями проектування реляційної бази даних є: повнота відображення всіх необхідних даних в базі даних; виключення надмірності даних; визначення оптимального числа збережених в базі даних відношень; нормалізація відношень бази даних для вирішення проблем, пов'язаних з оновленням та видаленням даних.

Визначення 1.20. Нормалізація відношення – це розбиття (або декомпозиція) відношення (таблиці) на інші відношення (2 або більше), що володіють кращими властивостями при додаванні, зміні або видаленні даних. Декомпозиція повинна бути оборотною, тобто виконуватися без втрат інформації.

Визначення 1.21. Відношення знаходиться в першій нормальній формі (1НФ) тоді і тільки тоді, коли всі домени, що використовуються, містять тільки скалярні значення.

Визначення 1.22. Відношення знаходиться в другій нормальній формі (2НФ) тоді і тільки тоді, коли воно знаходиться в першій нормальній формі, і кожен неключових атрибут неприводимо залежить від первинного ключа.

Визначення 1.23. Відношення знаходиться в третій нормальній формі (3НФ) тоді і тільки тоді, коли воно знаходиться в другій нормальній формі, і кожен неключовий атрибут нетранзитивно залежить від первинного ключа.

Визначення 1.24. Відношення знаходиться в нормальній формі Бойса–Кодда (НФБК) тоді і тільки тоді, коли кожна нетривіальна і неприводима зліва функціональна залежність має потенційний ключ в ролі детермінанта.

Визначення 1.25. Відношення знаходиться в нормальній формі Бойса–Кодда тоді і тільки тоді, коли детермінанти є потенційними ключами.

Приклад 1.1. Спроекувати базу даних контактів людини. Людина характеризується прізвищем, ім'ям, по батькові (ПІБ), датою народження, індивідуальним податковим номером (ІПН), контактними даними (телефон мобільний, домашній, робочий, пошта, скайп і т. д.). Контакти людини можуть змінюватись або додаватись нові, але поява нового типу контакту не повинна призводити до зміни структури бази даних.

Нехай перша таблиця, яка буде зберігати інформацію про людину, має назву PEOPLE та містить такі атрибути (поля): ID_People (ідентифікатор людини або первинний ключ відношення), FIO (прізвище, ім'я, по батькові), Date_B (дата народження), IPN (індивідуальним податковим номером). Введемо другу таблицю TYPE_CONTACT для збереження різноманітних типів контактних даних (телефон мобільний, домашній, робочий, пошта, скайп і т. д.) з такими атрибутами (полями): ID_TYPE_CONTACT (ідентифікатор типу контакту або первинний ключ відношення), NAME_CONTACT (назва відповідного типу контактних даних). Оскільки в однієї людини може бути декілька різних видів контактів (телефон, пошта, скайп), а також один і той самий вид контакту може бути у багатьох людей (мобільний телефон є у різних людей), ми приходимо до того, що в нашому прикладі з'являється відношення типу «багато–до–багатьох» між людьми (таблицями PEOPLE) та типами контактних даних (таблиця TYPE_CONTACT). Для реалізації такого типу відношення додаємо допоміжну проміжну таблицю CONTACT, яка містить атрибути ID_Contact (первинний ключ відношення або ідентифікатор контакту), ID_TYPE_CONTACT (ідентифікатор типу контакту), ID_People (ідентифікатор людини) та конкретне значення контакту VALUE_CONTACT (номер телефону, ім'я поштової скриньки і т. д.). На рис. 1.1 показана схема бази даних з урахуванням зовнішніх ключів для розглянутого прикладу.

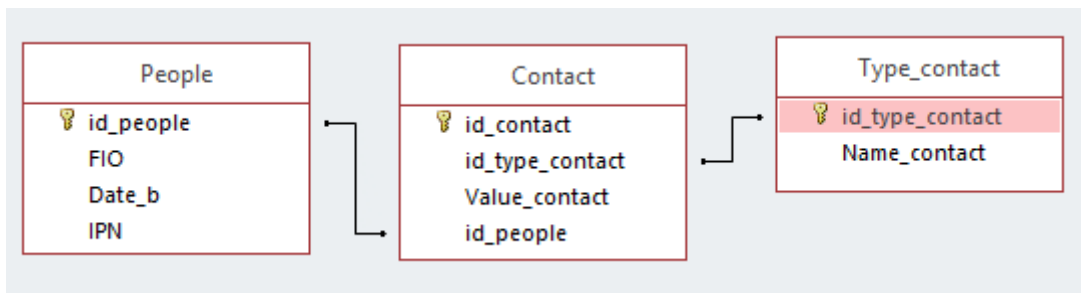


Рис. 1.1 – Схема бази даних контактів людини

1.2. Задачі для самостійного розв’язання

Задача 1.1. Спроекувати структуру бази даних для задачі обліку мешканців студмістечка

Студмістечко складається з декількох будівель (гуртожитків), в яких мешкають студенти, аспіранти та інші категорії мешканців. Місткість кожної кімнати визначається її типом. Вартість проживання визначається будівлею (гуртожитком), типом кімнати, категорією мешканця та кількістю мешканців в кімнаті. Мешканці характеризуються прізвищем, ім’ям, по батькові, датою народження, індивідуальним податковим номером, адресою, паспортними даними, підрозділом. Потрібно забезпечити можливість розрахунку планової вартості сплати за проживання. Для кожного мешканця гуртожитка потрібно зберігати історію періодів проживання в конкретній кімнаті та вартість проживання.

Навести ескіз інтерфейсу програмного забезпечення.

Задача 1.2. Спроекувати структуру бази даних для задачі обліку книг домашньої бібліотеки

Кожна книга містить такі відомості: назва, автори, видавництво, рік випуску, тираж. У книги може бути декілька авторів, а один автор може написати декілька книг. Одна й та сама книга може бути надрукована різними видавництвами. Для кожної книги потрібно передбачити коментар, який заповнюється власником бібліотеки.

Список читачів, які користуються бібліотекою, фіксований. Для кожного читача потрібно зберігати контактну інформацію (домашній телефон, мобільні телефони, електронна адреса та інше). Можуть з'являтися нові види контактів, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

Необхідно зберігати інформацію про те, хто і на який термін брав книги, коли повернув (чи не прострочений термін повернення). Також необхідно зберігати всю історію користування бібліотекою.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.3. Спроекувати базу даних для ведення журналу обліку успішності та відвідуваності навчальних занять академічною групою

Існує декілька груп студентів. Один студент може навчатись тільки в одній групі. Журнал пов'язаний з конкретною групою, яка складається з певної кількості студентів. Студент характеризується прізвищем, ім'ям, по батькові, адресою, телефоном, датою народження, курсом і номером групи.

Група вивчає конкретні дисципліни, які читаються конкретними викладачами. Викладач характеризується прізвищем, ім'ям, по батькові та датою народження. Одна й та сама дисципліна може читатись різними викладачами в різних групах. Для однієї групи дисципліна читається одним викладачем.

Необхідно зберігати інформацію про назви тем кожної дисципліни та їхню послідовність, проведені заняття, пропуски, отримані оцінки. Заняття проводиться в певний день на конкретній парі з відповідного предмета для кожної групи. Оцінки зберігаються окремим довідником, щоб під час зміни шкали оцінювання не змінювалась структура бази даних.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.4. Спроекувати структуру бази даних для задачі формування розкладу навчальних занять

Викладачі, які характеризуються прізвищем, ім'ям, по батькові, адресою, телефоном та датою народження, проводять заняття з певних дисциплін. Один викладач проводить заняття з декількох дисциплін, а одну дисципліну можуть читати декілька викладачів. Деякі заняття проводяться в комп'ютерних аудиторіях. Список аудиторій, які характеризуються номером та доступні для проведення занять, фіксований. Необхідно відрізнити звичайні та комп'ютерні аудиторії.

Заняття можуть проводитись для потоку, групи або підгрупи відповідного курсу навчання. Організація довідника підрозділів повинна бути ієрархічною (факультет → спеціальність → група → підгрупа). Необхідно враховувати тиждень (верхній або нижній), номер пари за порядком (1, 2 і т. д.), а також виділяти лекційні заняття.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.5. Спроекувати структуру бази даних для задачі обліку проданих автомобілів та наданих послуг станцією технічного обслуговування

Всеукраїнська фірма займається продажем та технічним обслуговуванням автомобілів. Фірма обслуговує тільки ті автомобілі, які вона продала. В різних містах України існують філіали СТО. Автомобіль, який був куплений в цій фірмі, може пройти ТО в будь-якому місті, де є її філіал. Необхідно зберігати інформацію про продаж, всі ТО для кожного автомобіля та СТО, на яких воно було зроблене.

В одному місті може бути декілька філіалів. Довідник міст виділити в окрему таблицю. Філіал займається продажем автомобілів, але не кожен філіал надає послуги ТО. Фірму цікавить повний перелік всіх філіалів по Україні.

Власник автомобіля характеризується прізвищем, ім'ям, по батькові, датою народження, коментарями менеджерів. Для кожного власника

автомобіля необхідно зберігати контактну інформацію (телефон домашній, мобільні телефони, електронна адреса та інше). Можуть з'являтися нові види контактів, але поява нового виду не повинна призводити до зміни структури бази даних. Власник може купити декілька автомобілів. Автомобіль може мати декілька власників (після перереєстрації). Обов'язково потрібно зберігати дату зміни власника.

Автомобіль характеризується серійним номером, кольором, моделлю (2107, 2101, Golf, Camry і т. д.), маркою (ВАЗ, ГАЗ, Nissan, BMW і т. д.). Марка автомобіля залежить від моделі.

Факт проходження ТО конкретним автомобілем характеризується датою та філіалом, а також інформацією (в текстовому вигляді) про причину проходження ТО.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.6. Спроекувати структуру бази даних для задачі обліку виданих абонементів відвідування басейну та їхньої оплати

Кожному відвідувачу видається абонемент або тимчасовий талон на одноразове відвідування басейну (характеризується номером та датою видачі). Абонементи можуть бути оформлені на кількість відвідувань, період відвідування або з обмеженням на кількість відвідувань протягом певного проміжку часу.

Відвідувачі характеризуються прізвищем, ім'ям, по батькові, адресою проживання, датою народження. Для кожного відвідувача необхідно зберігати контактну інформацію (телефон домашній, мобільні телефони, електронна адреса та інше). Можуть з'являтися нові види контактів, але поява нового виду не повинна призводити до зміни структури бази даних.

Сплата за послуги може здійснюватися готівковим або безготівковим розрахунком. Вид оплати зробити окремим довідником. Оплата здійснюється за весь абонемент в цілому та характеризується датою оплати.

Для кожного абонементу потрібно враховувати фактичне відвідування басейну (дата та час).

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.7. Спроекувати структуру бази даних для задачі обліку інформації про контрагентів

Контрагент містить такі реквізити: ЄДРПОУ (цифровий 10-значний код), назву контрагента, податковий номер, дату видачі податкового номера, банківські реквізити (може бути декілька). Банківські реквізити складаються з МФО (6-значний код банку) та розрахункового рахунку (цифровий код з 14 символів).

Довідник банків формується загальним списком по всій Україні. Банк характеризується номером МФО, назвою та містом, де він знаходиться. Контрагент в одному банку може мати декілька розрахункових або валютних рахунків. Для кожної окремої валюти відкривається окремий рахунок.

Необхідно зберігати контакти контрагентів (факс, телефон, електронна адреса та інше). Види контактів можуть з'являться нові, але поява нового контакту не повинна призводити до зміни структури бази даних. Також потрібно мати інформацію про те, чий саме контакт зберігається в базі даних (директора, головного бухгалтера, менеджера і т. д.).

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.8. Спроекувати структуру бази даних для задачі реєстрації клієнтів ветеринарної лікарні

Існують планові заходи (вакцинація, огляд і т. д.) для кожного виду тварин. Власники клієнтів ветеринарної лікарні характеризуються прізвищем, ім'ям, по батькові, адресою проживання, датою народження. Для кожного хазяїна необхідно зберігати контактну інформацію (телефон

домашній, мобільний, електронна пошта і т. д.). Можуть з'являтися нові види контактів, але поява нового виду не повинна призводити до зміни структури бази даних.

Тварини характеризуються видом, кличкою, датою реєстрації в ветеринарній лікарні, датою народження, примітками (заповнює співробітник). У кожної тварини є тільки один господар. Необхідно зберігати інформацію про фактичні звернення в лікарню (дата, час, причина звернення).

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.9. Спроекувати структуру бази даних для задачі обліку замовлень у фотоательє

Фотоательє характеризується пунктами прийому / видачі замовлень, назвою та адресою цих пунктів. Види наданих послуг (друк фотографій, обробка, оцифровка зображень та інше) однакові для всіх пунктів.

Замовник характеризується прізвищем, ім'ям, по батькові, датою народження, адресою, статусом (VIP, звичайний) і контактними даними різних видів. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

Виконавець характеризується прізвищем, ім'ям, по батькові, датою народження, адресою, посадою та контактними даними різних видів. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

Фотографії можуть бути різного розміру (тобто одна й та сама фотографія може бути надрукована різного розміру) і з різним терміном виконання роботи (термін залежить від розміру фотографії). Замовлення характеризується замовником, виконавцем, датою замовлення, видом послуг та кількістю фотографій. Для друку фотографія або завантажується або вказується адреса (URL) до файлу.

Загальна вартість визначається типом послуги, статусом клієнта, розміром фотографії (для друку), кількістю фотографій / послуг. Необхідно також зберігати інформацію про терміни виконання і зміни вартості послуг.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.10. Спроекувати структуру бази даних для задачі обліку інформації фільмів для відеопрокату

Фільм характеризується назвою, жанром, акторами, режисером, роком випуску, рейтингом. Для кожного фільму необхідно передбачити коментар, який заповнюється продавцем.

Для кожного покупця необхідно зберігати прізвище, ім'я, по батькові та контактну інформацію (телефон домашній, мобільний, електронні адреси та інше). Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

На носії (вони можуть бути різних типів – CD, DVD та інше) можуть зберігатись декілька фільмів. Необхідно зберігати інформацію про те, хто та на який термін брав носії з фільмами, а також, коли повернув (чи не прострочений термін повернення). Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.11. Спроекувати базу даних товарів медичного призначення

Аптека характеризується назвою, містом, адресою, телефоном або іншим видом контакту. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

На продаж надходять як медикаменти, так і медична техніка (тонометри, інгалятори та інше), а також інші медичні товари. Товар характеризується виробником, терміном придатності, обов'язковістю продажу за рецептом, призначенням ліків (для діабетиків, для хворих на серце та інше). Виробник характеризується назвою підприємства, містом та адресою, телефоном або іншим типом контакту.

Потрібно зберігати дату та час продажу товару, ким відпущений товар, кількість, ціна та вартість, назву товару, в якій аптеці був проданий, за яким рецептом (ким він виписаний). Товар поставляється в аптеку у відповідний день та час, його приймає співробітник аптеки. Потрібно зберігати перелік отриманого товару, його кількість, ціну та вартість. Довідник товарів повинен бути організований ієрархічно. В аптеці працюють співробітники, які характеризуються прізвищем, ім'ям, по батькові, датою народження, контактними даними та посадою.

Необхідно мати можливість розрахунку наявної кількості товарів в кожній аптеці.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.12. Спроекувати базу даних реєстрації багатофункціональних пристроїв (БФП) та принтерів університету

Кожен пристрій характеризується виробником, моделлю, типом (принтер або БФП), типом друку (струменевий, лазерний та інше), кольором (чорно-білий, кольоровий), датою виробництва, датою поставки до університету, інвентарним номером, а також інформацією про місцезнаходження та матеріально відповідальною особою. Університет має навчальні корпуси, кожен з яких складається з кімнат. На кожному поверсі є свої кімнати з окремою нумерацією.

Інвентарний номер повинен складатися з 8 символів. Модель БФП або принтера визначається виробником.

Співробітник характеризується прізвищем, ім'ям, по батькові, датою народження та контактними даними різних видів. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних. Один співробітник може бути матеріально відповідальною особою за декілька пристроїв.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.13. Спроекувати структуру баз даних для задачі обліку продажів продукції косметичної фірми

Замовлення характеризуються датою замовлення, загальною сумою та покупцем. Покупець характеризується прізвищем, ім'ям, по батькові, датою народження та місцем проживання (потрібно зробити окремий довідник міст). Для кожного покупця необхідно зберігати контактну інформацію (телефон домашній, мобільний, електронна пошта та інше). Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

В одному замовленні можуть бути присутні декілька позицій товарів та вказана їхня кількість, ціна за одиницю та загальна вартість позиції в замовленні. Довідник товарів повинен бути організований ієрархічно.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.14. Спроекувати структуру бази даних для задачі обліку продуктів на продовольчому складі

Довідник продуктів потрібно організувати ієрархічно. Продукти розрізняють за типом (м'ясні, молочні та інше) та одиницями виміру (кг, літри, пачки та інше), вартістю.

Постачальник характеризується назвою, кодом ЄДРПОУ, юридичною адресою та контактними даними різних видів. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних.

Продукти приймає комірник, який характеризується прізвищем, ім'ям, по батькові та датою народження. Поставки продуктів характеризуються датою та часом, номером, складським приміщенням та комірником, який приймає продукти, позиціями товарів та їхньою кількістю, вартістю і терміном зберігання кожного виду товару.

Відпуск товарів зі складу виконується комірником на основі документу-запиту, який заповнений та підписаний директором складу. Вимоги до документу характеризуються номером, датою, загальною сумою, позиціями товарів із зазначенням кількості та вартості. Товар відпускається відповідальній особі, інформація про яку вноситься в базу даних (прізвище, ім'я, по батькові та паспортні дані).

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.15. Спроекувати структуру бази даних для задачі обліку проведення спортивних заходів

Спортивні заходи характеризуються видом спорту, типом (змагання, олімпіада та інше), датою та часом проведення, командами-учасниками, сумою призового фонду, коментарями.

Потрібно передбачити, що для кожної команди визначається кінцеве зайняте місце (результат командних змагань). Команди можуть бути групові або одиночні.

Команда характеризується назвою, кількістю гравців, містом (місто реалізувати окремим довідником). Гравець характеризується прізвищем, ім'ям, по батькові, номером в команді, датою народження та контактними даними різних видів. Види контактів можуть з'являтися нові, але поява нового виду контакту не повинна призводити до зміни структури бази даних. Один гравець може входити до декількох команд.

Необхідно зберігати інформацію про суддів (жюрі) для спортивного змагання, а також знати головного суддю. Суддя може бути гравцем деякої команди. Суддя характеризується прізвищем, ім'ям, по батькові, датою народження та контактними даними різного виду. Види контактів можуть з'являться нові, але поява нового виду не повинна призводити до зміни структури бази даних.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.16. Спроекувати базу даних для задачі обліку документів деканату

В університеті є певна кількість факультетів. На кожному факультеті є свій деканат. Документи бувають вхідні та вихідні, різних типів (запит, скарга, наказ тощо), мають відповідального виконавця з числа співробітників деканату. Тип документу визначає максимальний термін відповіді або реагування.

Для кожного документу необхідно зберігати його номер та дату надходження, дату створення та короткий зміст. Кожен документ має ключові слова. Їх може бути декілька для одного документа.

Необхідно зберігати інформацію про співробітників деканату, які підписують той чи інший документ. Співробітників, які підписали документ, може бути декілька. Кожен співробітник характеризується прізвищем, ім'ям, по батькові та посадою.

Документ може бути пов'язаний зі співробітниками факультету або студентами. Необхідно зберігати інформацію про всіх пов'язаних осіб. Співробітники факультету характеризуються прізвищем, ім'ям, по батькові, датою народження, посадою, підрозділом / кафедрою. Довідник підрозділів організований ієрархічно. Студенти факультету характеризуються прізвищем, ім'ям, по батькові, датою народження, курсом, групою та спеціальністю.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.17. Спроекувати базу даних для задачі обліку автомобілів на автостоянці

Автомобіль характеризується реєстраційним номером, кольором, моделлю (2107, 2101 тощо) та маркою (ВАЗ, ГАЗ, BMW тощо). Модель автомобіля визначається його маркою.

Необхідно зберігати інформацію про власників автомобілів, які характеризуються прізвищем, ім'ям, по батькові, датою народження та контактними даними різних видів. Види контактів можуть з'являтися нові,

але поява нового виду не повинна призводити до зміни структури бази даних. Один власник може мати декілька автомобілів.

На автостоянці існує декілька тарифів, які залежать від термінів зберігання (добовий, місячний, квартальний та інше) або кількості заїздів (один, три, десять і т. д.). Необхідно зберігати інформацію як про актуальну вартість абонементів / талонів, так і про зміну тарифів.

Абонемент оформлюється власнику на кожен автомобіль окремо та характеризується номером абонементу, датою видачі та вартістю, паркомісцем (всі місця на стоянці пронумеровані). Також необхідно зберігати інформацію про дати заїзду та виїзду автомобіля.

Навести ескізи інтерфейсу програмного забезпечення.

Задача 1.18. Спроекувати базу даних для задачі обліку інформації про замовлення в магазині комп'ютерних комплектуючих

Комплектуючі характеризуються типом (процесор, материнська плата тощо), моделлю (AMD FX 8320 і т. д.), описом характеристик та ціною.

Покупець характеризується прізвищем, ім'ям, по батькові та контактними даними. Види контактів можуть додаватись нові, але поява нового виду не повинна призводити до зміни структури бази даних.

Оплата за товар може бути здійснена готівкою або безготівковим розрахунком. Вид оплати оформлюється окремим довідником. Один покупець може зробити декілька замовлень, і один продавець може зробити декілька продажів.

Необхідно в замовленні зберігати дату та час покупки, загальну вартість замовлення, тип, модель, термін дії гарантії та ціну комплектуючих, їхню кількість, вид оплати, прізвище, ім'я, по батькові покупця та прізвище, ім'я, по батькові продавця.

Навести ескізи інтерфейсу програмного забезпечення.

РОЗДІЛ 2. ОПЕРАЦІЇ РЕЛЯЦІЙНОЇ АЛГЕБРИ

2.1. Теоретичні відомості

Обов'язковою частиною реляційної моделі, яка використовується при проектуванні структур баз даних, є оператори. Реляційна алгебра складається з набору операторів, які використовують відношення в ролі операндів, і результатом виконання яких також є відношення.

Реляційна алгебра складається з восьми операторів, які поділені на дві групи по чотири оператори в кожній:

- традиційні операції: об'єднання, перетин, різниця, декартовий добуток;
- спеціальні операції: вибірка, проекція, з'єднання та ділення.

Визначення 2.1. Два відношення сумісні за типом, якщо в них ідентичні заголовки, тобто виконані такі дві вимоги: кожне з відношень має одну й ту ж множину імен атрибутів, та відповідні атрибути визначені на одному й тому ж домені.

Визначення 2.2. Об'єднанням двох сумісних за типом відношень A та B ($A \text{ UNION } B$) називається відношення з тим самим заголовком, як у відношеннях A та B , та тілом, яке складається з множини всіх кортежів, які належать A або B .

Визначення 2.3. Перетином двох сумісних за типом відношень A та B ($A \text{ INTERSECT } B$) називають відношення з тим самим заголовком, як у відношеннях A та B , та тілом, яке складається з множини всіх кортежів, які належать одночасно обом відношенням A та B .

Визначення 2.4. Різницею двох сумісних за типом відношень A та B ($A \text{ MINUS } B$) називається відношення з тим самим заголовком, як і в відношеннях A та B , та тілом, яке складається з множини всіх кортежів, які належать відношенню A та не належать відношенню B .

Визначення 2.5. Декартовим добутком двох відношень A та B ($A \text{ TIMES } B$), де A та B не мають спільних імен атрибутів, визначається як

відношення із заголовком, який представляє собою зчеплення двох заголовків початкових відношень A та B та тілом, яке складається з множини всіх кортежів, таких що представляє собою зчеплення кортежу, яке належить відношенню A та кортежу, яке належить відношенню B .

Визначення 2.6. Θ -вибіркою з відношення A за атрибутами X та Y (A WHERE $X \Theta Y$) називається відношення, яке має той самий заголовок, що і відношення A та тіло, яке містить множини всіх кортежів відношення A , для яких перевірка умови дає значення істини.

Атрибути X і Y повинні бути визначені на одному і тому ж домені, а оператор Θ повинен мати сенс для цього домену.

Визначення 2.7. Проекцією відношення A за атрибутами X, Y, \dots, Z , де кожен з атрибутів належить відношенню A ($A [X, Y, \dots, Z]$), називається відношення із заголовком $\{X, Y, \dots, Z\}$ і тілом, що містить множини всіх кортежів $\{X: x, Y: y, \dots, Z: z\}$, таких що у відношенні A значення атрибута X дорівнює x , атрибута Y дорівнює y, \dots , атрибута Z дорівнює z . Таким чином, за допомогою оператора проекції отримуємо вертикальну підмножину цього відношення.

Нехай відношення A і B мають заголовки $\{X_1, X_2, \dots, X_m, Y_1, Y_2, \dots, Y_n\}$ та $\{Y_1, Y_2, \dots, Y_n, Z_1, Z_2, \dots, Z_p\}$ відповідно, тобто атрибути Y_1, Y_2, \dots, Y_n (і тільки вони) – спільні для двох відношень. X_1, X_2, \dots, X_m – решта атрибутів відношення A , Z_1, Z_2, \dots, Z_p – решта атрибутів відношення B . Припустимо, що атрибути з однаковими іменами визначені на одному й тому ж домені. Розглянемо вирази $\{X_1, X_2, \dots, X_m\}$, $\{Y_1, Y_2, \dots, Y_n\}$ і $\{Z_1, Z_2, \dots, Z_p\}$ як три складові атрибута X, Y і Z відповідно.

Визначення 2.8. Природнім поєднанням відношень $A \{X, Y\}$ та $B \{Y, Z\}$ (A JOIN B) називається відношення із заголовком $\{X, Y, Z\}$ та тілом, яке містить множини всіх кортежів $\{X: x, Y: y, Z: z\}$ таких, для яких у відношенні A значення атрибута X дорівнює x , а атрибута Y дорівнює y , та у відношенні B значення атрибута Y дорівнює y , атрибута Z дорівнює z .

Θ -поєднанням відношення А за атрибутом Х з відношенням В за атрибутом Y ($A \text{ TIMES } B \text{ WHERE } X \Theta Y$), де відношення А і В не мають спільних імен атрибутів, називається відношення з тим самим заголовком, що і при декартовому добутку відношень А та В та тілом, яке містить множину кортежів таких, що належить цьому декартовому добутку та обчислення умови $X \Theta Y$ дає значення істини для цього кортежу.

Нехай відношення А і В мають заголовки $\{X_1, X_2, \dots, X_m, Y_1, Y_2, \dots, Y_n\}$ та $\{Y_1, Y_2, \dots, Y_n\}$ відповідно, тобто атрибути Y_1, Y_2, \dots, Y_n – спільні для двох відношень, відношення А має додаткові атрибути X_1, X_2, \dots, X_m , а відношення В не має додаткових атрибутів (Відношення А і В представляють відповідно ділене і дільник). Припустимо також, що атрибути з однаковими іменами визначені на одному й тому ж домені. Нехай вираження $\{X_1, X_2, \dots, X_m\}$ та $\{Y_1, Y_2, \dots, Y_n\}$ позначають два атрибути Х і Y відповідно.

Визначення 2.9. Діленням відношень А $\{X, Y\}$ на В $\{Y\}$ ($A \text{ DIVIDEBY } B$) називається відношення із заголовком $\{X\}$ та тілом, яке містить множину всіх кортежів $\{X:x\}$ таких, що існує кортеж $\{X:x, Y:y\}$, який належить відношенню А для всіх кортежів $\{Y:y\}$, які належать відношенню В.

Приклад 2.1. Задані відношення R1 та R2. Знайти об'єднання цих відношень.

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
QW	30	111
24	20	113
Q8	10	R9

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
24	30	111
QW	30	111
74	20	117

Відношення R1 та R2 сумісні за типом (в них ідентичні заголовки, тобто кожне з відношень має одну й ту ж множину імен атрибутів, та

відповідні атрибути визначені на одному й тому ж домені), тому можна визначити операцію об'єднання цих відношень.

R1 UNION R2		
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>Char(25)</i>
QW	30	111
24	20	113
Q8	10	R9
24	30	111
74	20	117

Приклад 2.2. Задані відношення R1 та R2 (приклад 2.1). Знайти перетин цих відношень.

Відношення R1 та R2 сумісні за типом (в них ідентичні заголовки, тобто кожне з відношень має одну й ту ж множину імен атрибутів та відповідні атрибути визначені на одному й тому ж домені), тому можна визначити операцію перетину цих відношень.

R1 INTERSECT R2		
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>Char(25)</i>
QW	30	111

Приклад 2.3. Задані відношення R1 та R2 (приклад 2.1). Знайти різницю цих відношень.

Відношення R1 та R2 сумісні за типом (в них ідентичні заголовки, тобто кожне з відношень має одну й ту ж множину імен атрибутів, та відповідні атрибути визначені на одному й тому ж домені), тому можливо визначити операцію різниці цих відношень.

R1 MINUS R2		
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>Char(25)</i>
24	20	113
Q8	10	R9

Приклад 2.4. Задані відношення R1 та R2. Визначити декартовий добуток цих відношень.

R1	
Field1 <i>Char(5)</i>	Field2 <i>int</i>
H1	11
R2	22
U3	33

R2		
Field3 <i>Char(5)</i>	Field4 <i>int</i>	Field5 <i>Char(25)</i>
W1	11	W04
A1	22	C03
H1	33	X03
R3	44	L01

R1 TIMES R2				
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>Char(5)</i>	Field4 <i>int</i>	Field5 <i>Char(25)</i>
H1	11	W1	11	W04
H1	11	A1	22	C03
H1	11	H1	33	X03
H1	11	R3	44	L01
R2	22	W1	11	W04
R2	22	A1	22	C03
R2	22	H1	33	X03
R2	22	R3	44	L01
U3	33	W1	11	W04
U3	33	A1	22	C03
U3	33	H1	33	X03
U3	33	R3	44	L01

Приклад 2.5. Для відношення R1 знайти R1 WHERE Field1 = Field3.

R1				
Field1	Field2	Field3	Field4	Field5
<i>Char(5)</i>	<i>int</i>	<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
H1	11	R2	11	W04
R2	22	R1	22	C03
U3	33	U3	33	X03
E3	33	E3	44	L01

Атрибути Field1, Field3 визначені на одному домені, операція « \Rightarrow » має сенс, тому визначимо R1 WHERE Field1 = Field3

R1 WHERE Field1 = Field3				
Field1	Field2	Field3	Field4	Field5
<i>Char(5)</i>	<i>int</i>	<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
U3	33	U3	33	X03
E3	33	E3	44	L01

Приклад 2.6. Задані відношення R1 та R2. Визначити природне поєднання цих відношень.

R1	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
H1	11
R2	22
U3	33

R2		
Field2	Field3	Field4
<i>int</i>	<i>int</i>	<i>Char(25)</i>
11	101	W04
11	201	C03
33	301	X03
44	401	L01

R1 JOIN R2			
Field1	Field2	Field3	Field4
<i>Char(5)</i>	<i>int</i>	<i>Char(5)</i>	<i>int</i>
H1	11	101	W04
H1	11	201	C03
U3	33	301	X03

Приклад 2.7. Задані відношення R1 та R2. Визначити ділення цих відношень.

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>int</i>
H1	101	10
H1	101	20
H1	101	30
R2	202	10
R2	202	20
R2	202	30
R2	202	40
U3	33	10
U3	33	20
U3	33	40
T5	77	10

R2
Field3
<i>int</i>
10
20
30

R1 DIVIDEBY R2	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
H1	101
R2	202

2.2. Задачі для самостійного розв'язання

Операція A INTERSECT B

Завдання 2.1. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення перетину цих відношень (R1 INTERSECT R2). Знайдіть його, якщо це можливо.

А)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	111
24	20	113
Q8	10	R9

R2		
Field4	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
24	20	113
54	30	111
74	20	117

Б)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
H1	12	X03
R2	40	C05
U3	15	G1
H1	12	F3

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	80	W4
A1	20	C03
H1	12	X03
R2	40	L1

В)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
B1	20	SX3
R2	30	CS5
Q3	14	GS1
R1	11	F3w

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R1	11	W4
U3	14	GS1
B1	20	X03
R2	30	CS5

Г)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	30	D4
T4	20	R3
R1	11	F3
Q8	10	R9

R2		
Field1	Field5	Field6
<i>Char(5)</i>	<i>Char(5)</i>	<i>Char(5)</i>
R2	30	R3
T4	20	R3
Q8	80	C5
R1	10	S7

Операція A UNION B

Завдання 2.2. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення об'єднання цих відношень (R1 UNION R2). Знайдіть його, якщо це можливо.

А)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	111
24	20	113
Q8	10	R9

R2		
Field4	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
24	20	113
54	30	111
74	20	117

Б)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
H1	12	X03
R2	40	C05
U3	15	G1

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
U3	15	G1
A1	20	C03
H1	12	X03

B)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
B1	20	SX3
R2	30	CS5
Q3	14	GS1

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	30	CS5
U3	14	GS1
B1	20	X03

Г)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	30	D4
T4	20	R3
R1	11	F3

R2		
Field1	Field5	Field6
<i>Char(5)</i>	<i>Char(5)</i>	<i>Char(5)</i>
R2	30	R3
T4	20	R3
Q8	80	C5

Операція A MINUS B

Завдання 2.3. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення різниці цих відношень (R1 MINUS R2). Знайдіть його, якщо це можливо.

A)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	111
24	20	113
Q8	10	R9

R2		
Field4	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
24	20	113
54	30	111
74	20	117

Б)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
H1	12	X03
R2	40	C05
U3	15	G1
E3	33	H1

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
U3	15	G1
A1	20	C03
H1	12	X03
T1	33	H1

В)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
B1	20	SX3
R2	30	CS5
Q3	14	GS1
T3	74	S1

R2		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	30	CS5
U3	14	GS1
B1	20	X03
Q1	70	S1

Г)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
R2	30	D4
T4	20	R3
R1	11	F3

R2		
Field1	Field5	Field6
<i>Char(5)</i>	<i>Char(5)</i>	<i>Char(5)</i>
R2	30	R3
T4	20	R3
Q8	80	C5

Операція А TIMES В

Завдання 2.4. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення декартового добутку цих відношень (R1 TIMES R2). Знайдіть його, якщо це можливо.

A)

R1	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
54	30
24	20
Q8	10

R2	
Field1	Field2
<i>int</i>	<i>int</i>
24	20
54	30
74	20

B)

R1	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
54	30
24	20
Q8	10

R2	
Field3	Field4
<i>int</i>	<i>int</i>
24	20
54	30

B)

R1	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
R2	30
U4	20
Q8	10

R2	
Field3	Field4
<i>Char(5)</i>	<i>int</i>
R2	30
U3	14

Γ)

R1	
Field1	Field2
<i>Char(5)</i>	<i>int</i>
R2	30
T4	20
R1	11

R2	
Field3	Field2
<i>int</i>	<i>int</i>
54	30
74	20
84	11

Операція A WHERE X Θ Y

Завдання 2.5. Задане відношення R1. Обґрунтуйте можливість / неможливість обчислення θ -вибірки з відношення R1 за атрибутами X та Y (R1 WHERE Field2>Field3). Знайдіть його, якщо це можливо.

R1			
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>Logic</i>	Field4 <i>Int</i>
U4	30	. T .	120
W4	20	. F .	220
Q8	10	. T .	320
R4	30	. T .	430

Завдання 2.6. Задане відношення R1. Обґрунтуйте можливість / неможливість обчислення θ -вибірки з відношення R1 за атрибутами X та Y (R1 WHERE (Field2+Field3)>Field4). Знайдіть його, якщо це можливо.

R1			
Field1 <i>Char(5)</i>	Field2 <i>int</i>	Field3 <i>int</i>	Field4 <i>Int</i>
U4	30	90	120
W4	20	300	220
Q8	10	10	320
R4	30	900	430

Завдання 2.7. Задане відношення R1. Обґрунтуйте можливість / неможливість обчислення θ -вибірки з відношення A за атрибутами X та Y відношення R1 (R1 WHERE (Field2-Field3)>Field4). Знайдіть його, якщо це можливо.

R1			
Field1	Field2	Field3	Field4
<i>Char(5)</i>	<i>int</i>	<i>int</i>	<i>Int</i>
U4	150	30	120
W4	20	300	220
Q8	900	10	320
R4	500	50	430

Завдання 2.8. Задане відношення R1. Обґрунтуйте можливість / неможливість обчислення θ -вибірки з відношення R1 за атрибутами X та Y відношення R1 (R1 WHERE Field1>Field3). Знайдіть його, якщо це можливо.

R1			
Field1	Field2	Field3	Field4
<i>Char(5)</i>	<i>int</i>	<i>Logic</i>	<i>int</i>
T4	30	.T.	12
W1	20	.F.	20
N8	10	.T.	32
R4	30	.T.	43

Операція A JOIN B

Завдання 2.9. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення природного з'єднання цих відношень (R1 JOIN R2). Знайдіть його, якщо це можливо.

A)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	111
24	20	113
Q8	10	R9

R2		
Field4	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
24	20	113
54	30	111
74	20	117

Б)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	11
24	20	13
Q8	10	R9
U8	70	Q9

R2		
Field2	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
30	20	113
30	30	111
10	20	117
90	20	17

В)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	11
24	20	13
Q8	10	R9
U8	70	Q9

R2		
Field3	Field5	Field6
<i>Char(25)</i>	<i>int</i>	<i>int</i>
11	20	113
11	30	111
10	20	117
R9	20	17

Г)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
54	30	111
24	20	113
Q8	10	R9

R2		
Field4	Field5	Field6
<i>int</i>	<i>int</i>	<i>int</i>
24	20	113
54	30	111
74	20	117

Операція A TIMES B WHERE X Θ Y

Завдання 2.10. Задані відношення R1 та R2. Обґрунтуйте можливість / неможливість обчислення θ -поєднання відношення R1 за

атрибутом Field1 з відношенням R2 за атрибутом Field4 цих відношень (A TIMES B WHERE Field1=Field4). Знайдіть його, якщо це можливо.

A)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>Char(25)</i>
H1	12	X03
R2	40	C05
U3	15	G1
H1	12	F3

R2		
Field1	Field4	Field5
<i>Char(5)</i>	<i>Char(5)</i>	<i>Char(25)</i>
H1	H1	W4
R2	R2	C03
H1	U3	W3
R2	W1	L1

B)

R1		
Field1	Field2	Field3
<i>int</i>	<i>int</i>	<i>Char(25)</i>
15	12	X03
25	40	C05
50	15	G1
75	12	F3

R2		
Field4	Field5	Field6
<i>int</i>	<i>Char(5)</i>	<i>Char(25)</i>
50	H1	W4
70	R2	C03
15	U3	W3
50	W1	L1

B)

R1		
Field1	Field2	Field3
<i>Char(25)</i>	<i>int</i>	<i>Char(25)</i>
W5	15	X3
U5	24	C5
U5	13	G1
E5	82	F3

R2		
Field4	Field5	Field6
<i>Char(25)</i>	<i>Char(5)</i>	<i>Char(25)</i>
U5	H1	W4
W5	R2	C3
P5	U3	W3
Q5	W1	L1

Γ)

R1		
Field1	Field2	Field3
<i>Char(5)</i>	<i>int</i>	<i>int</i>
H1	12	20
R2	40	30
U3	15	40
H1	12	50

R2		
Field1	Field5	Field6
<i>Char(5)</i>	<i>Char(25)</i>	<i>int</i>
R2	80	20
A1	20	40
H1	12	50
R2	40	70

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Томас Конноли, Каролин Бегг. Базы данных. Проектирование, реализация и сопровождение. Теория и практика, 3-е изд. М.: Издательский дом «Вильямс», 2003.
2. Дейт К. Введение в системы баз данных, 8-е изд. М.: Издательский дом «Вильямс», 2006.
3. Гектор Гарсиа-Молина, Джеффри Ульман, Дженифер Уидом. Системы баз данных. Полный курс. М.: Издательский дом «Вильямс», 2003.
4. Алексей Ковязин, Сергей Востриков. Мир InterBase. Архитектура, администрирование и разработка приложений баз данных в InterBase/Firebird/Yaffil. М.: Кудиц-Образ, 2005.
5. Хелен Борри. Firebird: руководство разработчика баз данных. Санкт-Петербург: БХВ-Петербург, 2006.

Допоміжна література

1. Системы баз данных: Экон. прил.: учеб. пособие / В. Н. Андриенко, Я. Г. Берсуцкий, В. Г. Скобелев, А. С. Томяковский; Донецкий гос. унт. Донецк: ДонГУ, 1999. 213 с.

Інформаційні ресурси в Інтернет

1. Документація Майкрософт для користувачів, розробників та ІТ-спеціалістів. URL: <https://docs.microsoft.com/ru-ru/>
2. Документація по базам даних та СУБД Firebird. URL: <http://www.ibase.ru/start/>
3. Документація по базі даних SQL Azure. URL: <https://docs.microsoft.com/ru-ru/azure/sql-database/>
4. Офіційна документація по СУБД Firebird. URL: <https://firebirdsql.org/en/firebird-rdbms/>
5. <https://moodle.donnu.edu.ua/course/view.php?id=68>