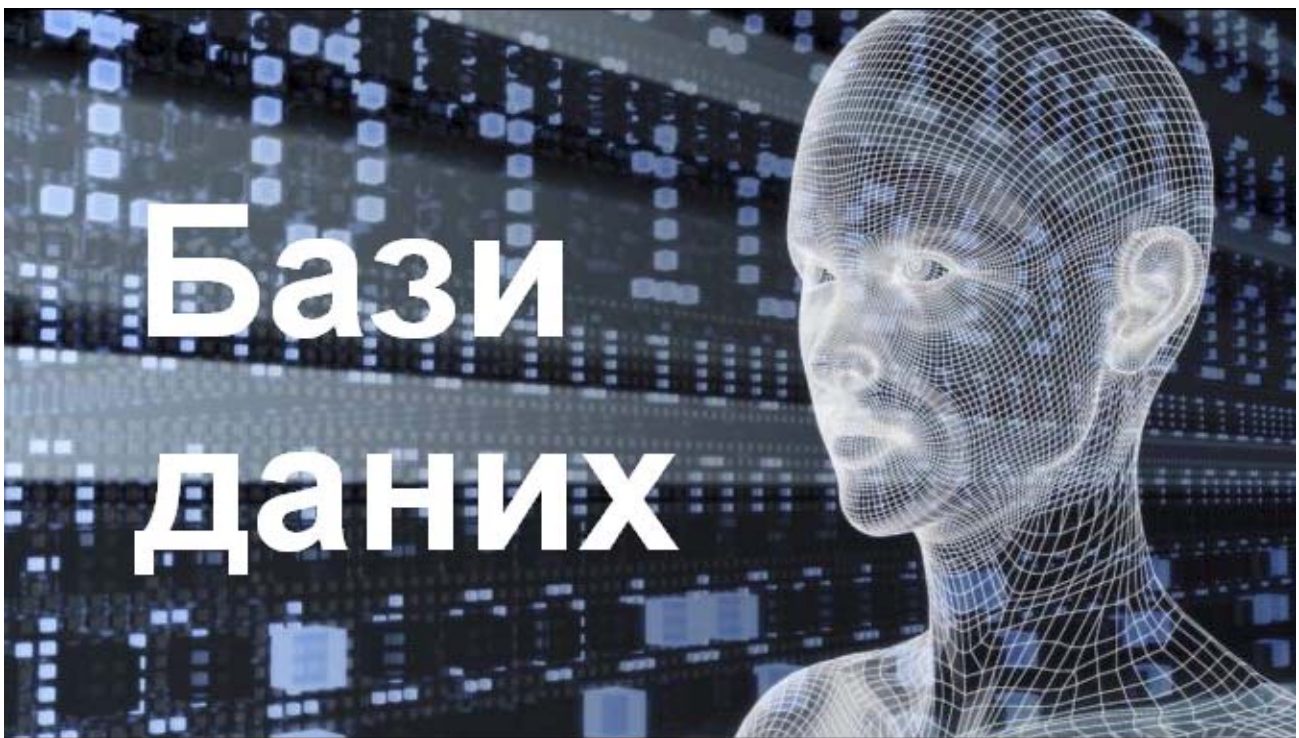


О. Н. Романюк, О. В. Романюк, А. В. Денисюк

ПРАКТИКУМ
для самостійної роботи студентів
з навчальної дисципліни «Бази даних»



Міністерство освіти і науки України
Вінницький національний технічний університет

ПРАКТИКУМ
для самостійної роботи студентів
з навчальної дисципліни «Бази даних»

Вінниця
ВНТУ
2022

УДК 004.65

Р 69

Рекомендовано до видання Вченою радою Вінницького національного технічного університету Міністерства освіти і науки України (протокол № 7 від 31.03.2022 р..)

Рецензенти:

С. В. Павлов, доктор технічних наук, професор ВНТУ

О. М. Васілевський, доктор технічних наук, професор ВНТУ

Т. І. Коробейнікова, кандидат технічних наук, доцент НУ «Львівська політехніка»

Романюк, О. Н.

Р69 Практикум для самостійної роботи студентів з навчальної дисципліни «Бази даних» : електронний практикум комбінованого (локального та мережного) використання. [Електронний ресурс] / О. Н. Романюк, О. В. Романюк, А. В. Денисюк – Вінниця : ВНТУ, 2022. – 85 с.

Практикум рекомендується для самостійної роботи студентів з навчальної дисципліни «Бази даних» студентами, які навчаються за бакалаврською програмою зі спеціальності 121 – Інформаційні технології.

Практикум з дисципліни «Бази даних» розроблено на основі багаторічного досвіду викладання авторами дисципліни «Бази даних».

У практикуму наведено приклади розв'язання різних типів задач, які найбільш поширені в даній галузі.

Використання запропонованих навчально-методичних матеріалів сприятиме актуалізації процесу підготовки, індивідуалізації та диференціації освітнього процесу, запровадженню особистісного підходу в навчальній діяльності.

УДК 004.65

ЗМІСТ

1 ПРОЄКТУВАННЯ ТА НОРМАЛІЗАЦІЯ БАЗ ДАНИХ	4
2 МОВА ЗАПИТІВ SQL.....	39
3 ТЕСТОВІ ЗАВДАННЯ 1	56
4 ТЕСТОВІ ЗАВДАННЯ 2.....	64
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	80

1 ПРОЄКТУВАННЯ ТА НОРМАЛІЗАЦІЯ БАЗ ДАНИХ

1.1 Навести архітектуру системи клієнт-сервер

За технологією клієнт-сервер припускається, що, крім зберігання БД, центральний комп'ютер (сервер БД) має забезпечувати виконання основного обсягу обробки даних. При технології клієнт-сервер запит на виконання операції з даними (наприклад, звичайний вибір), який видає клієнт, змушує сервер виконувати пошук і вилучення даних. Отримані дані, але не файли, транспортуються по мережі від сервера до клієнта.

Архітектура клієнт-сервер (рис. 1.1), для якої призначені серверні СКБД, є, деякою мірою, поверненням до колишньої «мейнфреймової» моделі, заснованої на централізації збереження й обробки даних на одному виділеному комп'ютері, де функціонує спеціальна прикладна програма або сервіс, що називається сервером БД.

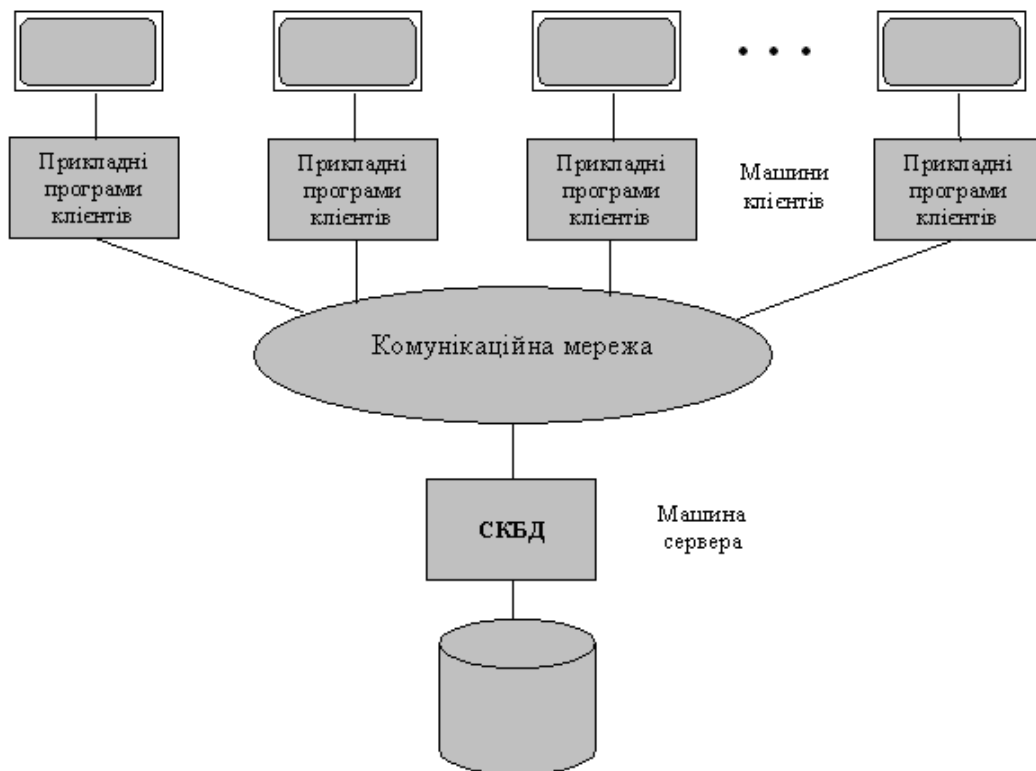


Рисунок 1.1 – Архітектура системи клієнт-сервер

Сервер БД відповідає за роботу з файлами БД, підтримку цілісності посилань, резервне копіювання, забезпечення авторизованого доступу до даних, протоколювання операцій і, звичайно, за виконання запитів користувача на вибір й модифікацію даних і метаданих (визначення інших об'єктів системи). Клієнтські прикладні програми, що є джерелами цих запитів, функціонують на персональних комп'ютерах у мережі. Можна

відзначити, що при використанні серверних СКБД виконання запитів виробляється самим сервером, тому клієнтські прикладні програми одержують від сервера тільки результати самого запиту і не вимагають передачі всього індексу чи всієї таблиці, що істотно знижує мережний трафік при обробці запитів.

1.2 Навести приклад ієрархічної структури даних: директор, заступник, керівники відділів, співробітники

На рис. 1.2 наведено приклад ієрархічної структури даних. Структура містить корінь і вузли.

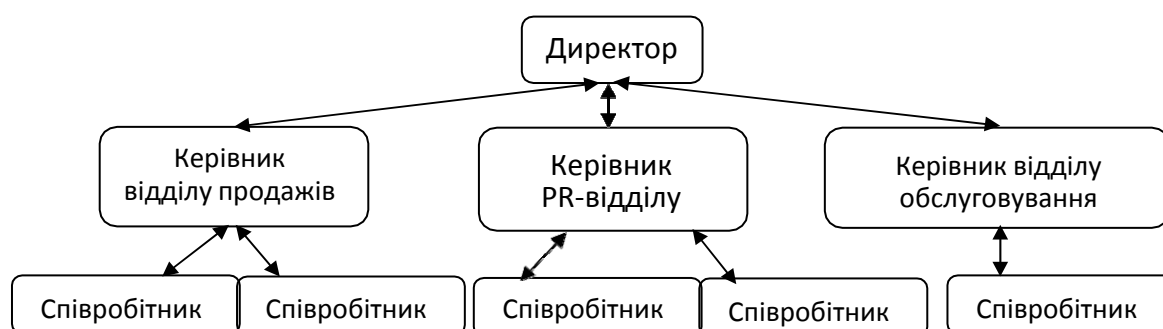


Рисунок 1.2 – Ієрархічна структура даних

1.3 Навести вимоги до банків даних

До основних вимог відносять:

- Адекватність інформації про стан предметної галузі. Інформація має повно та точно відображати об'єкти, їх властивості і відношення між ними.
- Надійність функціонування – одна з найважливіших вимог, які висуваються до будь-якої системи.
- Швидкодія і продуктивність. Ці дві близькі одна до одної вимоги відображають часові потреби користувачів. Перша з них визначається часом відповіді (реакції) системи на запит, який відраховується з моменту введення запиту до моменту початку видачі знайдених даних. Друга визначається кількістю запитів, які відпрацьовуються в одиницю часу.
- Простота і зручність використання. Ця вимога висувається до банку даних з боку всіх без винятку категорій користувачів, особливо кінцевих.
- Масовість використання. Сучасна інформаційна система має забезпечувати колективний доступ користувачів, при якому вони можуть одночасно і незалежно звертатися до баз даних для одержання необхідних даних.
- Захист інформації. Система має забезпечувати захист збережених у ній даних і програм як від випадкових спотворень і знищення, так і від навмисних, несанкціонованих дій користувачів.

- Можливість розширення. Архітектура системи має допускати розширення її можливостей шляхом модифікації або заміни існуючих програмних модулів, додаванням нових компонентів, а також шляхом реорганізації інформаційних масивів.

1.4 Навести приклад простої асоціації

В простій асоціації типу 1 (рис. 1.3) екземпляр елемента даних, від якого направлено зв'язок, ідентифікує один і лише один екземпляр елемента даних, до якого направлено зв'язок. Ця ідентифікація є унікальною й визначає функціональну залежність.

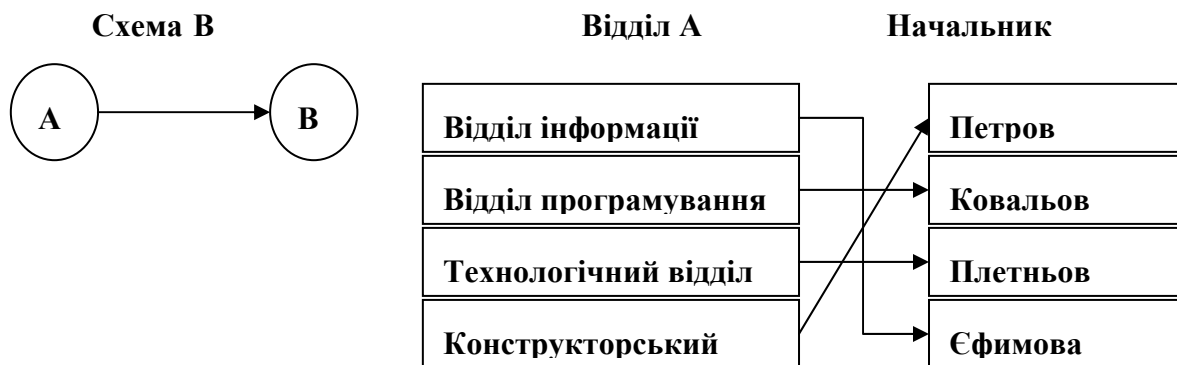


Рисунок 1.3 – Приклад простої асоціації типу 1

Навести приклад умовної асоціації типу С

В умовній асоціації типу С (рис. 1.4) для даного екземпляра елемента даних, від якого спрямований зв'язок, може не існувати відповідного екземпляра елемента даних, до якого спрямований зв'язок. Якщо він існує, то стосується єдиного екземпляра елемента даних. Наприклад, ПІБ РОБІТНИКА і ДАТА ЗВІЛЬНЕННЯ.



Рисунок 1.4 – Приклад умовної асоціації типу С

1.6 Навести приклад складної асоціації

У складній асоціації типу М (рис. 1.5) екземпляр елемента даних, від якого направлено зв'язок, ідентифікує деяке число екземплярів елементів даних, до яких направлено зв'язок. Ідентифікація є багатозначною залежністю і не обов'язково унікальною.

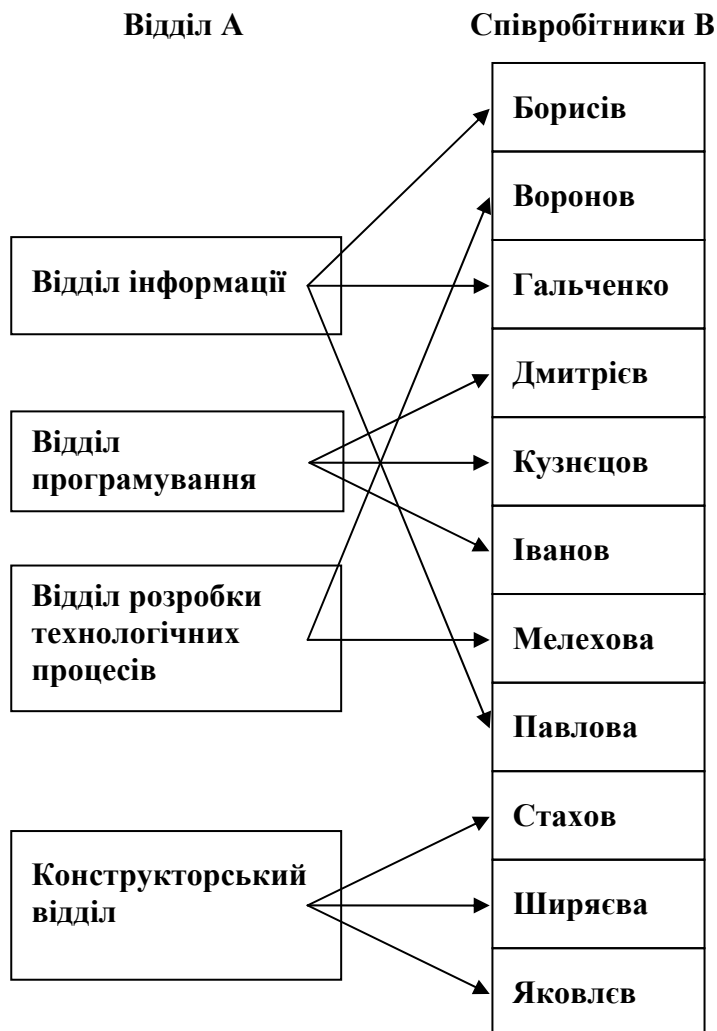


Рисунок 1.5 – Приклад складної асоціації типу М

1.7 Знайти декартовий добуток відношень R і F .

$$R = \begin{bmatrix} x a \\ y a \\ z a \end{bmatrix}, \quad F = \begin{bmatrix} 1 a 1 \\ 2 b 1 \end{bmatrix}$$

Дія декартового добутку – $R \times F$. Тут R і F можуть мати різні схеми. Ступінь відношення $R \times F$ дорівнює сумі ступенів відносин R і F . Ступінь відношення $R \times F$ дорівнює добутку ступенів відношення R і F . Нижче наведено приклад декартової операції.

$$R \times F = \begin{bmatrix} x & a & 1 & a & 1 \\ x & a & 2 & b & 1 \\ y & a & 1 & a & 1 \\ y & a & 2 & b & 1 \\ z & a & 1 & a & 1 \\ z & a & 2 & b & 1 \end{bmatrix}$$

1.8 Виконати операцію проєкції над відношенням СТУДЕНТ-УСПШНІСТЬ.

Таблиця 1.1 – СТУДЕНТ-УСПШНІСТЬ

ПІБ студента	Назва Дисципліни	Дата складання Екзамену	Оцінка
Іванов П. В.	Фізика	13.01.2000	4
Іванов П. В.	Фізика	14.06.2000	5
Іванов П. В.	Мат. аналіз	10.01.2000	3
Петров А. С.	Фізика	3.01.2000	3

Операція проєкції полягає у виділенні необхідних стовпців (доменів) з відношення.

Таблиця 1.2 – Приклад операції «проєкція»

ПІБ студента	Назва дисципліни	Оцінка
Іванов П. В.	Фізика	4
Іванов П. В.	Фізика	5
Іванов П. В.	Мат. аналіз	3
Петров А. С.	Фізика	3

1.9 Складіть з використання операції вибірки список співробітників, у яких табельний номер перевищує 80

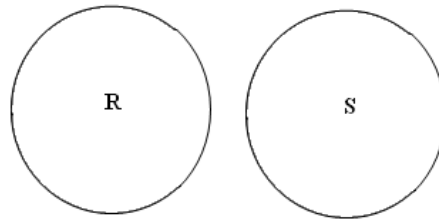
$$\sigma_{\text{ClockNumber} > 80}(\text{Librarians}).$$

Тут вихідним є відношення *Librarians*, а предикатом – вираз $\text{ClockNumber} > 80$. Операція вибірки визначає нове відношення, що містить тільки ті кортежі відношення *Librarians*, де значення атрибута *ClockNumber* перевищує 80 (табл. 1.3). Більш складні предикати можуть бути створені за допомогою логічних операторів *AND*, *OR* і *NOT*.

Таблиця 1.3 – Приклад операції «вибірки»

Code	Clock Number	FamilyName	Name	Patronymic	Pasport Code	Post
3	187	Іноземцева	Іванна	Модестівна	9	Ст. бібліотекар
4	83	Мальцева	Діана	Петрівна	12	Бібліотекар
6	100	Ставка	Лілія	Іванівна	7	Бібліотекар

1.10 Задано відношення R і S . Графічно зобразіть їх різницю



На рис. 1.6 показана графічна ілюстрація операції віднімання відношень

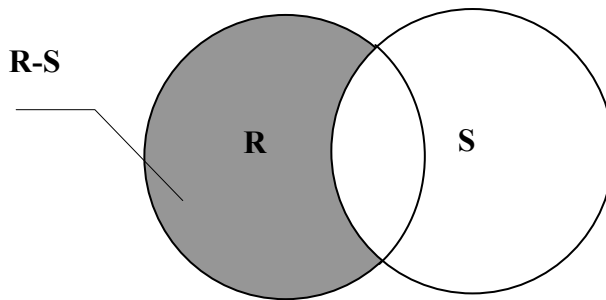


Рисунок 1.6 – Графічна ілюстрація операції віднімання відношень

1.11 Знайти відношення ВІДОМІСТЬ як декартовий добуток відношень СТУДЕНТИ та ЕКЗАМЕНИ

Таблиця 1.4 – СТУДЕНТИ

Студенти
Гаврилюк Піддубчак Скрипник Ящук

Таблиця 1.5 – ЕКЗАМЕНИ

Дисципліна	Дата	Оцінка
Математика	10.06.2022	
Фізика	15.06.2022	

Таблиця 1.6 – ВІДОМІСТЬ

Студенти	Дисципліна	Дата	Оцінка
Гаврилюк	Математика	10.06.2022	
Гаврилюк	Фізика	15.06.2022	
Піддубчак	Математика	10.06.2022	
Піддубчак	Фізика	15.06.2022	
Скрипник	Математика	10.06.2022	
Скипник	Фізика	15.06.2022	
Ящук	Математика	10.06.2022	
Ящук	Фізика	15.06.2022	

1.12 Знайти відношення $R=R1:R2$

Таблиця 1.7 – Відношення $R1$

Прізвище	Предмет	Оцінка
Литовко	Хімія	5
Литовко	Фізика	5
Петров	Хімія	4
Петров	Фізика	4
Сидоров	Хімія	5
Сидоров	Фізика	4

Таблиця 1.8 – Відношення $R2$

Предмет	Оцінка
Фізика	4

Відношення-ділене $R1$ має містити як підмножину множини атрибутів відношення дільника. Частка $R = R1:R2$ містить тільки ті атрибути діленого, яких немає в дільнику. У відношення частки внесені ті кортежі, декартовий добуток, яких із відношення-дільника міститься у відношенні діленого.

Таблиця 1.9 – Відношення $R = R1:R2$

Прізвище
Сидоров

1.13 Виконати селекцію відношення згідно з ознакою (вік>20)

Таблиця 1.10 – Відношення

ІМ'Я студента	Вік
Гаврилюк	21
Піддубчак	20
Скрипник	23
Ящук	19

В результаті отримаємо відношення як в табл. 1.11.

Таблиця 1.11 – Відношення після селекції

ПІБ студента	Вік
Гаврилюк	21
Піддубчак	20
Скрипник	23
Ящук	19

1.14 Виконаємо операцію ділення над відношеннями, які подані, відповідно, таблицями 1.12 і 1.13

Таблиця 1.12 – Екзаменаційна відомість

Студенти	Дисципліна	Дата	Оцінка
Гаврилюк	Математика	10.03.2000	3
Гаврилюк	Фізика	15.03.2000	5
Піддубчак	Математика	10.03.2000	4
Піддубчак	Фізика	15.03.2000	5
Скрипник	Математика	10.03.2000	5
Скрипник	Фізика	15.03.2000	4
Ящук	Математика	10.03.2000	4
Ящук	Фізика	15.03.2000	5

Таблиця 1.13 – Відношення-дільник

Дисципліна	Оцінка
Математика	4
Фізика	5

У результаті отримуємо частку (табл. 1.14).

Таблиця 1.14 – Відношення-частка

Студенти
Піддубчак
Ящук

1.15 Виконати проєкцію відношення

Таблиця 1.15 – Відношення

Номер студента	Ім'я	Група	Дата народження
1	Кулаков Петро	2	23.04.1990
2	Васильєв Іван	1	11.05.1991
4	Мишкіна Ірина	2	23.09.1991

Результатом проєкції є відношення, що містить всі кортежі після видалення з них деяких атрибутів. В цьому випадку кортежі-результати називаються *підкортежами*.

Виберемо з таблиці лише імена та дати народження: Отримаємо відношення

Таблиця 1.16 – Відношення після проєкції

Ім'я	Дата народження
Кулаков Петро	23.04.1990
Васильєв Іван	11.05.1991
Мишкіна Ірина	23.09.1991

1.16 Виконати ділення відношень R1 на R2

Результатом ділення двох відношень (бінарного і унарного) є відношення

R1:

A	X
A	Y
B	Z
B	X
C	Y

R2:

X
Y

Відношення, що містить всі значення атрибута першого бінарного відношення, які відповідають всім значенням унарного відношення.

У результаті отримаємо відношення

R:

A

1.17 Нехай дано таке відношення Students. Привести його до першої нормальної форми (1НФ)

Таблиця 1.17 – Відношення Students

StudentID	Name
1	Козаков Петро Володимирович
2	Васильєв Іван Аркадійович
3	Шишкіна Дарина Сергіївна

Відношення знаходиться в 1НФ тоді і тільки тоді, коли всі використовувані домени містять тільки скалярні значення, тобто значення всіх полів відношення мають бути неподільні.

Поле **Name** містить одночасно прізвище, ім'я та по батькові. Це поле можна розділити на три, тоді отримаємо відношення в 1НФ:

Таблиця 1.18 – Відношення Students в 1НФ

StudentID	LastName	FirstName	MiddleName
1	Козаков	Петро	Володимирович
2	Васильєв	Іван	Аркадійович
3	Шишкіна	Дарина	Сергіївна

Потрібно мати на увазі, що значення полів мають бути **логічно** неподільними, тому що прізвище складається з окремих букв, але такий поділ не матиме сенсу для відношення *Students*.

1.18 Привести відношення (Таблиця 1.19) до другої нормальної форми (2НФ)

Відношення знаходиться в 2НФ тоді і тільки тоді, коли воно знаходиться в 1НФ і кожен неключовий атрибут повністю залежить від первинного ключа.

Таблиця 1.19 – Вихідне відношення

StudentID	LastName	FirstName	MiddleName	Course	Score
1	Козаков	Петро	Володимирович	інформатика	5
2	Васильєв	Іван	Аркадійович	математика	4
4	Шишкіна	Дарина	Сергіївна	математика	5
2	Васильєв	Іван	Аркадійович	інформатика	3

Первинним ключем тут буде $\{\text{StudentID}, \text{Course}\}$, тоді проаналізуємо ФЗ цього відношення, де детермінантою є первинний ключ, а в правій частині – неключовий атрибут. Розгляньмо таку ФЗ

$$\{\text{StudentID}, \text{Course}\} \rightarrow \{\text{LastName}\}$$

Очевидно, що вона є привідною, тому що прізвище залежить тільки від **StudentID**, а значить існує ФЗ: $\{\text{StudentID}\} \rightarrow \{\text{LastName}\}$.

Отже, дане відношення не знаходиться у 2НФ. Виконаємо декомпозицію вихідного відношення на такі два.

Таблиця 1.20 – Відношення Students

StudentID	LastName	FirstName	MiddleName
1	Козаков	Петро	Володимирович
2	Васильєв	Іван	Аркадійович
4	Шишкіна	Дарина	Сергіївна

Таблиця 1.21 – Відношення Scores

StudentID	Course	Score
1	Інформатика	5
2	Математика	4
4	Математика	5
2	Інформатика	3

1.19 Привести відношення (таблиця 1.22) до третьої нормальної форми (3НФ)

Таблиця 1.22 – Відношення Students

StudentID	LastName	FirstName	MiddleName	GroupID	Supervisor
1	Козаков	Петро	Володимирович	1	Царьов С. М.
2	Васильєв	Іван	Аркадійович	2	Пестов Д. Н.
4	Шишкіна	Дарина	Сергіївна	1	Царьов С. М.

Відношення знаходиться в 3НФ тоді і тільки тоді, коли воно знаходиться у 2НФ і кожен неключовий атрибут не є транзитивно залежним від первинного ключа (це означає, що у відношенні відсутні будь-які взаємні залежності). Оскільки існують ФЗ: **StudentID** \rightarrow **GroupID** і **GroupID** \rightarrow **Supervisor**, то атрибут **Supervisor** транзитивно залежить від первинного ключа **StudentID**. Відтак відношення не перебуває у 3НФ. Наведемо ставлення до 3НФ.

Таблиця 1.23 – Відношення Students

StudentID	LastName	FirstName	MiddleName	GroupID
1	Козаков	Петро	Володимирович	1
2	Васильєв	Іван	Аркадійович	2
4	Шишкіна	Дарина	Сергіївна	1

Таблиця 1.24 – Відношення Scores

GroupID	Supervisor
1	Царьов С. М.
2	Пестов Д. Н.

Третя нормальна форма дозволяє виправити ті аномалії, які були не виправлені під час переходу від 1НФ до 2НФ через транзитивні функціональні залежності.

1.20 Привести до третьої нормальної форми відношення

Таблиця 1.25 – Вихідне відношення

Ідентифікатор курсу	Рік	Лектор	Телефон
1	2020	Корін Г. А.	111-11-11
2	2019	Кавун А. Х.	222-22-22
2	2020	Кавун А. Х.	222-22-22
3	2019	Ревіна А. Б.	333-33-33

У ньому є дві базові функціональні залежності:

Ідентифікатор курсу, Рік → Викладач
Викладач → Телефон.

Через те, що **Телефон** лише транзитивно залежить від ключа, мають місце аномалії вставки.

Щоб позбутися транзитивних залежностей, виконаємо декомпозицію за останньою в кожному ланцюжку ФЗ залежністю; повторюватимемо таку операцію, поки всі ланцюжки залежностей не стануть довжиною 1.

В даному прикладі це єдиний ланцюжок ФЗ:

Ідентифікатор курсу, Рік → Викладач → Телефон

Виконавши декомпозицію за **Викладач → Телефон**, отримаємо

Таблиця 1.26 – Відношення Рік→Викладач→Телефон

Ідентифікатор курсу	Рік	Викладач	Телефон
1	2020	Корін Г. А.	111-11-11
2	2019	Кавун А. Х.	222-22-22
2	2020	Кавун А. Х.	222-22-22
3	2019	Ревіна А. Б.	333-33-33
3	2020	Чепурний А. И.	444-44-44

Таблиця 1.27 – Відношення Викладач→Телефон

Викладач	Телефон
Корін Г. А.	111-11-11
Кавун А. Х.	222-22-22
Ревіна А. Б.	333-33-33
Чепурний А. И.	444-44-44

Неважко помітити, що після приведення до 3НФ не залишається жодних «неявних» залежностей між атрибутами одного відношення. Завдяки цьому аномалії вставки, видалення та зміни більше не виявляються: незалежні один від одного безпосередньо неключові дані ніяк не впливають на можливості зберегти або оновити ту чи іншу інформацію.

1.21 Привести відношення до форми Бойса-Кодда

Таблиця 1.28 – Вихідне відношення

Номер клієнта	Дата співбесіди	Час співбесіди	Номер кімнати	Номер співробітника
С345	13.10.2003	13:00	103	А138
С355	13.10.2003	13:05	103	А136
С368	13.09.2003	13:00	102	А154
С366	13.09.2003	13:30	105	А307

Відношення знаходиться в формі Бойса-Кодда (BCNF), якщо воно знаходиться в 3НФ і немає залежності атрибутів первинного ключа від неключових атрибутів.

Ситуація, коли відношення буде в 3NF, але не в BCNF, виникає за умови, що відношення має два (або більше) можливі ключі, які є

складовими та мають загальний атрибут. Зауважимо, що на практиці така ситуація зустрічається досить рідко, для всіх інших відносин 3NF та BCNF еквівалентні. В результаті приведення до форми Бойса-Кодда отримуємо дві таблиці:

Таблиця 1.29 – Відношення 1

Номер клієнта	Дата співбесіди	Час співбесіди	Номер співробітника
C345	13.10.2003	13:00	A138
C355	13.10.2003	13:05	A136
C368	13.09.2003	13:00	A154

Таблиця 1.30 – Відношення 2

Дата співбесіди	Номер співробітника	Час співбесіди
13.10.2003	A138	13:00
13.10.2003	A136	13:05
13.09.2003	A154	13:00
13.09.2003	A307	13:30

1.22 Привести таблицю до нормальної форми Бойса Кодда

Таблиця 1.31 – Вихідне відношення

Проект	Напрямок	Куратор
1	Розробка	Панов В .В.
1	Бухгалтерія	Чанов В. В.
2	Розробка	Панов В .В.
2	Бухгалтерія	Петров П. П.
2	Реалізація	John Smith
3	Розробка	Андрів В. В.

У даному випадку маємо організацію, яка реалізує множину різних проектів. При цьому в кожному проекті робота ведеться за декількома функціональними напрямками, у кожному з яких є куратор. Співробітник може бути куратором лише у тому напрямку, у якому він спеціалізується, тобто, якщо співробітник програміст, він не може бути куратором в проекті, пов'язаному з бухгалтерією.

Припустимо, що потрібно зберігати інформацію про кураторів усіх проєктів у кожному напрямку.

У результаті реалізуємо таблицю, в якій первинний ключ є складовим «Проект + Напрямок», тому що в кожному проєкті є кілька напрямків роботи і тому, знаючи тільки проєкт, не можна визначити куратора напрямку, так само як, знаючи лише напрямок, не можна визначити куратора. Потрібно знати і проєкт, і напрямок, щоб визначити куратора цього напрямку у цьому проєкті.

Таблиця знаходиться у третій нормальній формі, тому що є первинний ключ, а неключовий стовпець залежить від усього ключа, а не від якоїсь його частини.

Але в даному випадку таблиця не знаходиться в нормальній формі Бойса-Кодда. Справа у тому, що, знаючи куратора, можна чітко визначити куратором якого напрямку він є, іншими словами, частина складового ключа, тобто, «Напрямок», залежить від неключового атрибута «Куратор».

Щоб привести цю таблицю до нормальної форми Бойса-Кодда, потрібно виконати декомпозицію цього відношення, тобто. розбити цю таблицю на кілька таблиць.

Таблиця 1.32 – Таблиця кураторів

Ідентифікатор куратора	ПІБ	Напрямок
1	Панов В .В.	Розробка
2	Чанов В. В.	Бухгалтерія
3	Петров П. П.	Бухгалтерія
4	John Smith	Реалізація
5	Андрів В. В.	Розробка

Таблиця 1.33 –Таблиця зв'язків кураторів і проєктів

Проект	Ідентифікатор куратора
1	1
1	2
2	1
2	3
2	4
3	5

Отже, у таблиці кураторів зберігається список кураторів та його спеціалізація, тобто напрямок, в якому він може бути куратором, а в таблиці зв'язку кураторів і проєктів відображається їх зв'язок.

1.23 Привести відношення до четвертої нормальної форми

Таблиця 1.34 – Вихідне відношення

Курс	Викладач	Аудиторія
SQL	Козак І. В.	101
SQL	Козак І. В.	203
SQL	Чабан Н. П.	305
SQL	Чабан Н. П.	407
Python	Марків В. П.	502
Python	Марків В. П.	305

Вимога четвертої нормальної форми (4NF) у тому, щоб у таблицях були відсутні нетривіальні багатозначні залежності.

Почнемо з того, що таблиця має мати як мінімум три стовпці, припустимо А, В і С, при цьому В і С між собою ніяк не пов'язані і не залежать один від одного, але окремо залежать від А, і для кожного значення А є множина значень В, і навіть множина значень С.

У такому разі багатозначна залежність позначається $A \twoheadrightarrow B$, $A \twoheadrightarrow C$.

Якщо подібна багатозначна залежність є у таблиці, вона не відповідає четвертій нормальній формі.

Нехай в якомусь закладі освіти є курси, де навчаються студенти, і викладачі, які читають ці курси, та аудиторії, в яких викладачі проводять заняття з курсів.

Таблиця 1.35 – Відношення курси

Ідентифікатор курсу	Назва курсу
1	SQL
2	Python
3	JavaScript

Таблиця 1.36 – Відношення Викладачі

Ідентифікатор викладача	ПІБ
1	Козак І. В.
2	Чабан Н. П.
3	Марків В. П.

Таблиця 1.37 – Відношення Аудиторії

Ідентифікатор аудиторії	Назва аудиторії
1	101
2	203
3	305
4	407
5	502

Той самий курс можуть викладати різні викладачі, і необов'язково в якійсь одній аудиторії, один раз курс може читатися в одній аудиторії, а в інший раз – в іншій аудиторії, наприклад, на курс записалося набагато менше студентів. Під цей потік можуть виділити аудиторію меншого розміру.

Також потрібно зазначити, що під кожен курс підходить лише певний набір аудиторій, наприклад, ті, що оснащені необхідним обладнанням, або ті, які мають відповідну місткість для цього курсу.

У закладі освіти, звичайно ж, постійно виникають питання зі складанням розкладу, проте для того, щоб його скласти, потрібно заздалегідь знати можливості цього закладу освіти. Іншими словами, які викладачі можуть викладати той чи інший курс, а також у яких аудиторіях той чи інший курс може читатись.

Для цього нам потрібно поєднати ці три сутності в одній таблиці. У результаті отримуємо таку таблицю.

Таблиця 1.38 – Інтегральне відношення

Курс	Викладач	Аудиторія
SQL	Козак І. В.	101
SQL	Козак І. В.	203
SQL	Чабан Н. П.	305
SQL	Чабан Н. П.	407
Python	Марків В. П.	502
Python	Марків В. П.	305

У даному випадку первинний ключ тут складається з усіх трьох стовпців, тому ця таблиця автоматично знаходиться у третій нормальній формі та нормальній формі Бойса-Кодда. Однак вона не знаходиться у четвертій нормальній формі, тому що тут є багатозначна залежність

Курс → Викладач

Курс → Аудиторія

Для кожного курсу у цій таблиці може бути кілька викладачів, а також кілька аудиторій.

При цьому потрібно врахувати, що викладачеві немає значення, в якій аудиторії читати лекцію, так само як і самій аудиторії немає значення, який викладач у ній працюватиме.

Іншими словами, ці два атрибути «Викладач» та «Аудиторія» ніяк не залежать один від одного, але вони окремо залежать від курсу.

Розглянемо деякі колізії.

Що буде, якщо, наприклад, викладач Козак І. В. звільнився. Потрібно буде видалити два рядки з цієї таблиці, але, видаливши ці рядки, видалимо всю інформацію і про аудиторії 101 і 203. Але вони насправді є і мають брати участь у плануванні розкладу. Це аномалія.

Чи інша ситуація – що буде, якщо курсу призначено викладача, але аудиторію ще не визначено? Або, навпаки, з аудиторією вже визначилися, а ось викладач ще невідомий.

Потрібно створити записи з NULL або зі значеннями за замовчуванням, і це також є аномалією.

Багатозначні залежності не можна незалежно редагувати. Іншими словами, щоб внести зміни до однієї залежності, неминуче мають задіяти іншу залежність. Тому головне правило четвертої нормальної форми звучить так: «у таблиці не має бути багатозначних залежностей».

Для цього виконують декомпозицію.

Таблиця 1.39 – Зв'язок курсів і викладачів

Курс	Викладач
SQL	Козак І. В.
SQL	Чабан Н. П.
Python	Марків В. П.

Таблиця 1.40 – Зв'язок курсів і аудиторій

Курс	Аудиторія
SQL	101
SQL	203
SQL	305
SQL	407
Python	502
Python	305

1.24 Визначити ключ відношення (ШИФР ГРУПИ, КІЛЬКІСТЬ СТУДЕНТІВ, ФАКУЛЬТЕТ, СТАРОСТА, КУРАТОР)

У відношенні можливими ключами можуть бути атрибути: ШИФР ГРУПИ, СТАРОСТА. А такі атрибути, як КІЛЬКІСТЬ СТУДЕНТІВ, ФАКУЛЬТЕТ не можуть бути ключами, оскільки різні студентські групи можуть мати однакову кількість студентів і на одному факультеті вчиться декілька груп.

1.25 Визначити ключ відношення (СТУДЕНТ-УСПІШІСТЬ)

Таблиця 1.41 – Відношення СТУДЕНТ-УСПІШІСТЬ

Прізвище, ім'я, по батькові студента	Назва дисципліни	Дата складання екзамену	Оцінка	Залік
Іванов П. В.	фізика	13.12.98	4	Залік
Іванов П. В.	фізика	17.12.98	5	Залік
Іванов П. В.	математичний аналіз	13.12.98	3	Залік
Петров А. С.	фізика	03.01.99	3	Залік

ПІБ СТУДЕНТА і ДАТА СКЛАДАННЯ ЕКЗАМЕНУ визначають кожний рядок даної таблиці. Якщо припустити, що можливе складання двох і більше екзаменів одним студентом в один день, тоді зазначені атрибути вже не будуть однозначно ідентифікувати кортеж, оскільки даним екземплярам атрибутів може відповідати в таких випадках два і більше кортежі.

В цьому випадку ПІБ СТУДЕНТА, ДАТА СКЛАДАННЯ ЕКЗАМЕНУ, НАЗВА ДИСЦИПЛІНИ будуть ключем.

1.26 Виконати об'єднання відношень, поданих таблицями

Таблиця 1.42 – ВИКЛАДАЧІ – ДИСЦИПЛІНИ 1

Викладачі	Дисципліни
Майданюк В. П.	Мікропроцесорні системи
Власюк В. Х.	Основи програмування
Романюк О. Н.	Засоби машинної графіки

Таблиця 1.43 – ВИКЛАДАЧІ – ДИСЦИПЛІНИ 2

Викладачі	Дисципліни
Арапов С. М.	Експертні системи
Войтко В. В.	Мережі ЕОМ
Обідник Д. Т.	Схемотехніка ЕОМ

Об'єднання відношень R і S (позначається $R \cup S$) являє собою множину кортежів, які належать R чи S або їм обом. Оператор об'єднання застосовується тільки до відношень однакової арності. Якщо в результаті об'єднання відношень мають місце однакові кортежі, то вони замінюються одним.

Виконаємо над ними операцію об'єднання. В результаті об'єднання відношень отримуємо підсумкове відношення, яке наведено в табл. 1.44.

Таблиця 1.44 – ВИКЛАДАЧІ – ДИСЦИПЛІНИ

Викладачі	Дисципліни
Арапов С. М.	Експертні системи
Войтко В. В.	Мережі ЕОМ
Обідник Д. Т.	Схемотехніка ЕОМ
Майданюк В. П.	Мікропроцесорні системи
Власюк В. Х.	Основи програмування
Романюк О. Н.	Засоби машинної графіки

1.27 Визначити нормальну форму, в якій знаходиться відношення **ФАКУЛЬТЕТ (НАЙМЕНУВАННЯ, ПІБ ДЕКАНА, ТЕЛЕФОН)**

Відношення **ФАКУЛЬТЕТ** задано в 2НФ, тому що у відношенні **ФАКУЛЬТЕТ** атрибут **ТЕЛЕФОН**, який не є основним, повністю залежить від будь-якого можливого ключа: **НАЙМЕНУВАННЯ, ПІБ ДЕКАНА**.

1.28 Виконати нормалізацію відношення **СТУДЕНТ – КУРС ПРОЄКТ (НОМЕР_ЗАЛІКОВОЇ_КНИЖКИ, КОД_ПРЕДМЕТА, ПРІЗВИЩЕ_СТУДЕНТА, НОМЕР_ГРУПИ, ВИКЛАДАЧ, ПРОЦЕНТ_ВИКОНАННЯ)**

Припустимо, що в одній групі можуть навчаються однофамільці. Тоді для цього відношення можливий тільки один ключ: **НОМЕР_ЗАЛІКОВОЇ_КНИЖКИ, КОД_ПРЕДМЕТА**. Виходячи з прийнятого припущення, атрибут **ПРІЗВИЩЕ_СТУДЕНТА** не входить у ключ. Тоді атрибут **НОМЕР_ЗАЛІКОВОЇ_КНИЖКИ** не визначається значенням атрибута **ПРІЗВИЩЕ_СТУДЕНТА**, тобто атрибути **ПРІЗВИЩЕ_СТУДЕНТА** і **НОМЕР_ГРУПИ** не є основними, але функціонально залежать від основного атрибута **НОМЕР_ЗАЛІКОВОЇ_КНИЖКИ**, що входить у складовий ключ. Залежності між атрибутами цього відношення показані на рис. 1.7.



Рисунок 1.7 – Функціональна залежність

Розщепивши вихідне відношення на два нових у другій нормальній формі, можна усунути надлишковість (рис. 1.8).

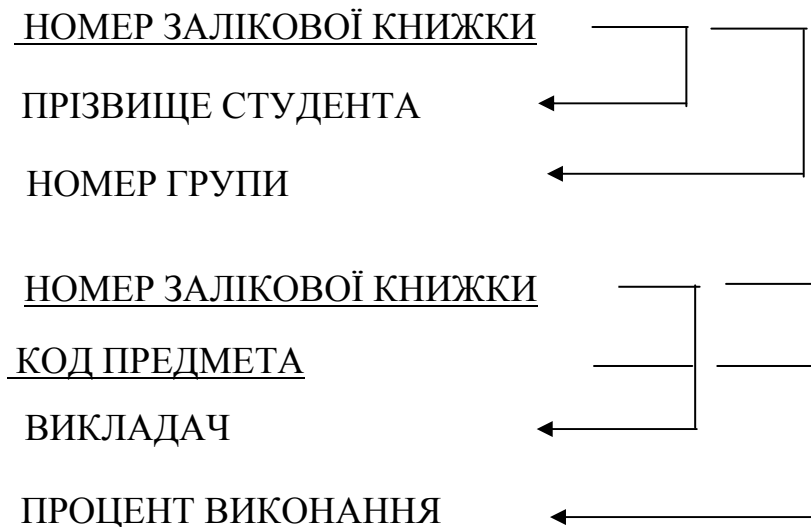


Рисунок 1.8 – Приклад усунення функціональної залежності

При виконанні цієї операції розбиття на два відношення враховано те, що атрибути, які функціонально залежать від одного основного атрибута, разом із ним утворять одне відношення з єдиним ключем НОМЕР_ЗАЛІКОВОЇ_КНИЖКИ, а інші атрибути, які функціонально повно залежать від складового ключа, залишено у вихідній схемі.

1.29 Виконати нормалізацію відношення: СПІВРОБІТНИКИ – ВІДДІЛИ – ПРОЄКТИ (СПІВРОБ_НОМЕР, СПІВРОБ_ЗАРП, ВІДДІЛ_НОМЕР, ПРО_НОМЕР, СПІВРОБ_ЗАВДАННЯ)

У відношенні використані скорочення: СПІВРОБ – співробітник, ЗАРП – зарплата, ПРО – проект.

Первинний ключ:
СПІВРОБ_НОМЕР, ПРО_НОМЕР.

Функціональні залежності:
СПІВРОБ_НОМЕР → СПІВРОБ_ЗАРП
СПІВРОБ_НОМЕР → ВІДДІЛ_НОМЕР
ВІДДІЛ_НОМЕР → СПІВРОБ_ЗАРП
СПІВРОБ_НОМЕР, ПРО_НОМЕР → СПІВРОБ_ЗАВДАННЯ.

Хоча первинним ключем є складовий атрибут СПІВРОБ_НОМЕР, ПРО_НОМЕР, атрибути СПІВРОБ_ЗАРП і ВІДДІЛ_НОМЕР

функціонально залежать від частини первинного ключа, тобто від атрибута СПВРОБ_НОМЕР. У результаті неможливо вставити у відношення СПВРОБІТНИКИ-ВІДДІЛИ-ПРОЄКТИ кортеж, що описує співробітника, який ще не виконує ніякого проєкта (первинний ключ не може містити невизначене значення). При видаленні кортежу не тільки руйнується зв'язок даного співробітника з даним проєктом, а й втрачається інформація про те, в якому відділі він працює. При переведенні співробітника в інший відділ потрібно модифікувати всі кортежі, які описують цього співробітника, інакше одержимо неузгоджений результат. Такі неприємні явища називаються аномаліями схеми відношення. Вони усуваються шляхом нормалізації.

Виконаємо декомпозицію відношення СПВРОБІТНИКИ-ВІДДІЛИ в два відношення СПВРОБІТНИКИ-ВІДДІЛИ і СПВРОБІТНИКИ-ПРОЄКТИ:

СПВРОБІТНИКИ-ВІДДІЛИ (СПВРОБ_НОМЕР, СПВРОБ_ЗАРП, ВІДДІЛ_НОМЕР)

Первинний ключ:
СПВРОБ_НОМЕР

Функціональні залежності:
СПВРОБ_НОМЕР -> СПВРОБ_ЗАРП
СПВРОБ_НОМЕР -> ВІДДІЛ_НОМЕР
ВІДДІЛ_НОМЕР -> СПВРОБ_ЗАРП

СПВРОБІТНИКИ-ПРОЄКТИ (СПВРОБ_НОМЕР, ПРО_НОМЕР, СПВРОБ_ЗАВДАННЯ)

Первинний ключ:
СПВРОБ_НОМЕР, ПРО_НОМЕР

Функціональна залежність:
СПВРОБ_НОМЕР, ПРО_НОМЕР -> СПВРОБ_ЗАВДАННЯ

Кожне з цих двох відношень знаходиться в 2НФ і в них усунуті відзначені вище аномалії.

1.30 Привести до третьої нормальної форми відношення ГУРТОЖИТОК (ПІБ_СТУДЕНТА, НОМЕР_ГРУПИ, НОМЕР_КІМНАТИ, СТАРОСТА_КІМНАТИ)

Відношення ГУРТОЖИТОК (ПІБ_СТУДЕНТА, НОМЕР_ГРУПИ, НОМЕР_КІМНАТИ, СТАРОСТА_КІМНАТИ) знаходиться в другій нормальній формі, але не в третій, тому що атрибут СТАРОСТА_КІМНАТИ залежить від атрибута НОМЕР_КІМНАТИ, який,

у свою чергу, залежить від атрибута ПІБ_ СТУДЕНТА і, отже, СТАРОСТА_КІМНАТИ транзитивно залежить від ПІБ_ СТУДЕНТА. Це відношення можна привести до необхідної форми шляхом його розщеплення на два:

СТУДЕНТ-ГУРТОЖИТОК (ПІБ_ СТУДЕНТА, НОМЕР_ ГРУПИ, НОМЕР_ КІМНАТИ)

та

КІМНАТА-ГУРТОЖИТОК (НОМЕР_ КІМНАТИ, СТАРОСТА_ КІМНАТИ).

1.31. Проаналізувати та нормалізувати відношення (ФІРМА, СКЛАД, ОБ'ЄМ).

Для даного відношення характерні такі аномалії:

1. Якщо в даний момент відсутня фірма, яка отримує товар зі складу, то в базу даних неможливо ввести інформацію про об'єм складу.

2. Якщо фірма перестає отримувати товар зі складу, то дані про склад та його об'єм не можна зберігати в базі даних.

3. Якщо об'єм складу змінився, то потрібно переглянути всі рядки відношення і змінити кортежі для форм, пов'язаних зі складом.

Причиною аномалій для даного відношення є наявність транзитивного зв'язку між атрибутами.

Для усунення аномалій розіб'ємо вихідне відношення на два:

ЗБЕРІГАННЯ (ФІРМА, СКЛАД) та ОБ'ЄМ (СКЛАД, ОБ'ЄМ).

1.32 Навести приклад індексно-послідовної організації даних

В індексно-послідовному методі доступу індексний файл завжди упорядкований за так званим первинним ключем. Первинний ключ – головний атрибут фізичного запису. За його значенням ідентифікується фізичний запис. Як правило, записи зберігаються в тій же логічній послідовності, що і індекс (звідси і назва індексно-послідовний метод), тому індекс має містити посилання не на кожний запис бази даних за ключем, а на групу записів в фізичному блоці за діапазоном ключів. Наприклад, якщо в блоці зберігається 10 записів, то для нього в індексному файлі буде одна стаття, а не десять, і розмір індексного файлу зменшиться в 10 разів.

Послідовна організація індексного файлу допускає індексацію його вмісту. Записи індексу групуються в блоки, які можна також індексувати. При роботі з більшими файлами така організація дозволяє покращити характеристики доступу.

На практиці ключі звичайно розподілені нерівномірно, через що попереднє закріплення значень ключів за блоками небажано, тому при створенні файлу даних виконується початкове завантаження в нього фізичних записів, упорядкованих за первинним ключем.

Найбільше або найменше значення ключа останнього запису, що міститься в блоці (більшість методів доступу забезпечує завантаження файлу даних за зростанням значення первинного ключа), заноситься в індексний файл (рис. 1.9).

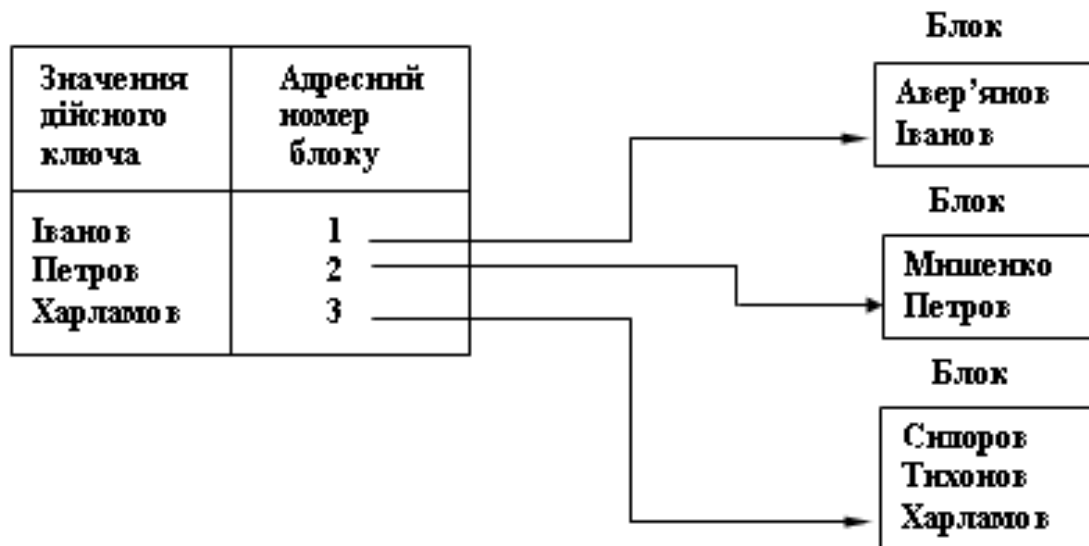


Рисунок 1.9 – Індексно-послідовний метод доступу

В цьому прикладі індексний файл і файл даних організовані послідовно. Індексний файл містить тільки максимальні значення первинних ключів записів кожного блоку. Поєднання кодів літер слова ТИХОНОВ більше в числовий інтерпретації поєднання кодів літер слова СИДОРОВ, але менше поєднання кодів літер слова ХАРЛАМОВ.

1.33 Побудувати діаграму ER-типів предметної області «База даних футбольного клубу», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Працівники (<ID_Працівника>, Ім'я_Працівника, Контакти, Робочий_час);
2. Посада (<Назва_посади>, Зарплата);
3. Вболівальник (<ID_Відвідувача>, Ім'я_вболівальника, Ціна_квитка);
4. Замовлення (<ID_Замовлення>, Назва_товару, Ціна_товару, Кількість_товару, Дата_відправлення, Адреса, Статус);
5. Відгуки (<Дата_відгуку>, Зміст_відгуку);
6. Матчі (<Дата_матчу>, Суперник, Місце_проведення, Результат, Кількість_глядачів);
7. Спонсори (<Назва_спонсора>, Сума, Тривалість_угоди, Опис_спонсора);
8. Травми (<Вид_травми>, Час_реабілітації, Ймовірність_рецидиву).

Діаграма ER-типів предметної області «База даних футбольного клубу» наведена на рисунку 1.10.

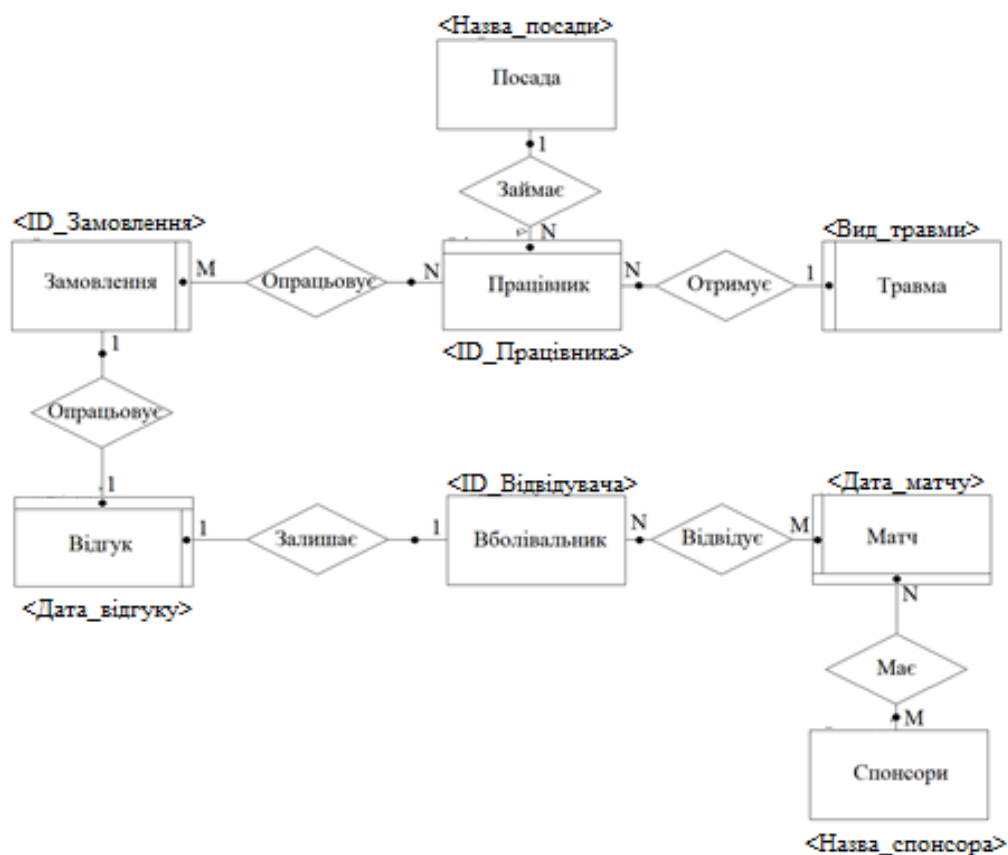


Рисунок 1.10 – Діаграма ER-типів предметної області «База даних футбольного клубу»

1.34 Побудувати діаграму ER-типів предметної області «База даних станції технічного обслуговування», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Працівник (<ID_Працівника>, ПІБ_Працівника, Номер_Працівника, Робочий_час);
2. Клієнт (<ID_Клієнта>, ПІБ_клієнта, Телефон_Клієнта, Email_клієнта);
3. Авто (<ID_Автомобіля>, Марка_Автомобіля, Модель_Автомобіля, Державний_Номер, Країна_Виробник, Рік_Випуску, Номер_Тех.паспорта, Колір, Номер_Двигуна);
4. Посада (<Назва_Посади>, Опис_Посади, Зарплата);
5. Замовлення (<Номер_Замовлення>, Опис_Замовлення, Вартість_Замовлення, Дата_Замовлення);
6. Робота (<ID_Роботи>, Дата_Виконання, Тривалість_Роботи, Найменування_Роботи).

Діаграма ER-типів предметної області «База даних станції технічного обслуговування» наведена на рисунку 1.11.

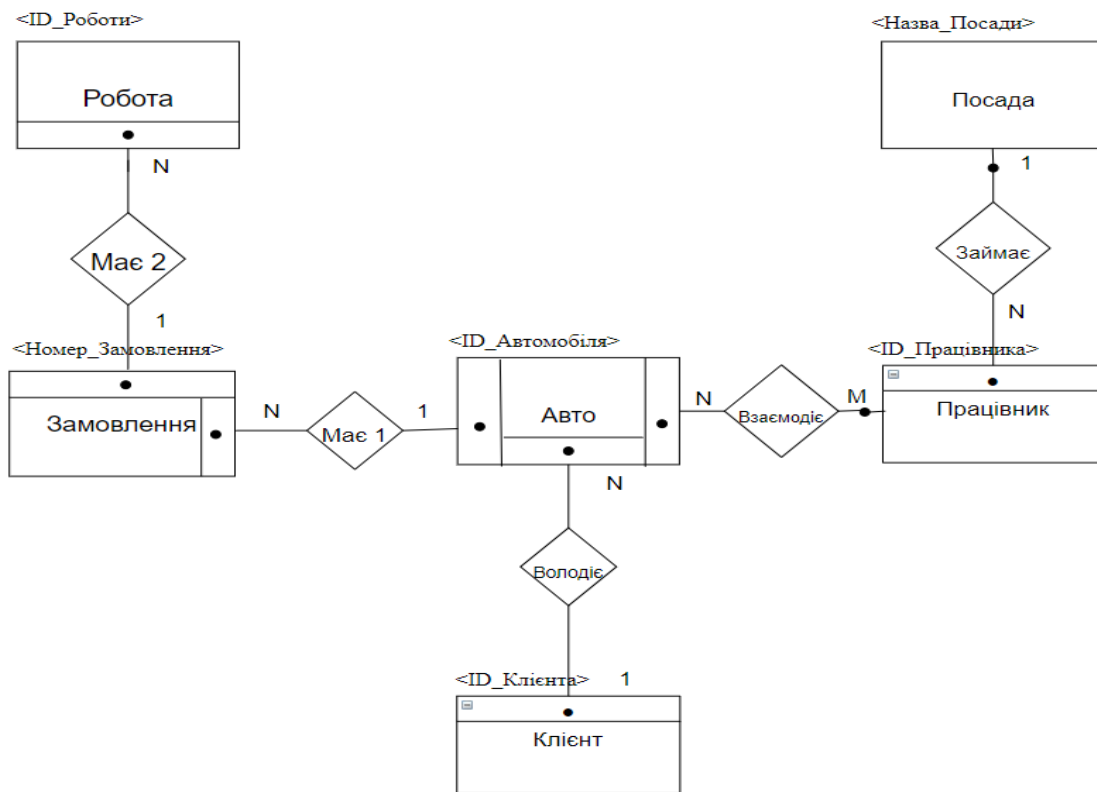


Рисунок 1.11 – Діаграма ER-типів предметної області «База даних станції технічного обслуговування»

1.35 Побудувати ER-діаграму екземплярів сутей Працівник і Посада, враховуючи, що клас належності є обов'язковим для суті Працівник та необов'язковим для суті Посада. Тип зв'язку N:1

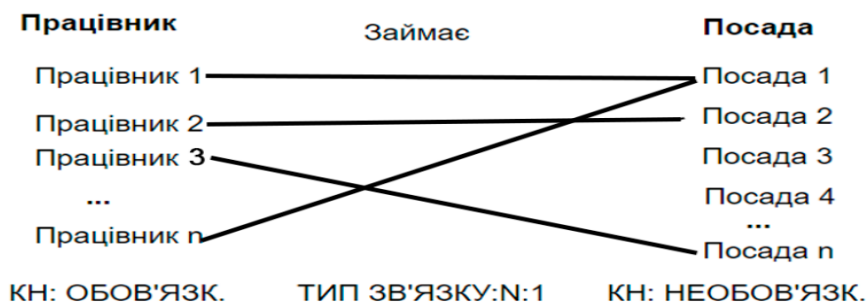


Рисунок 1.12 – ER-діаграма екземплярів сутей Працівник і Посада

1.36 Використовуючи діаграму функціональних залежностей (рис. 1.13), записати відношення, які характеризують предметну область «База даних станції технічного обслуговування»

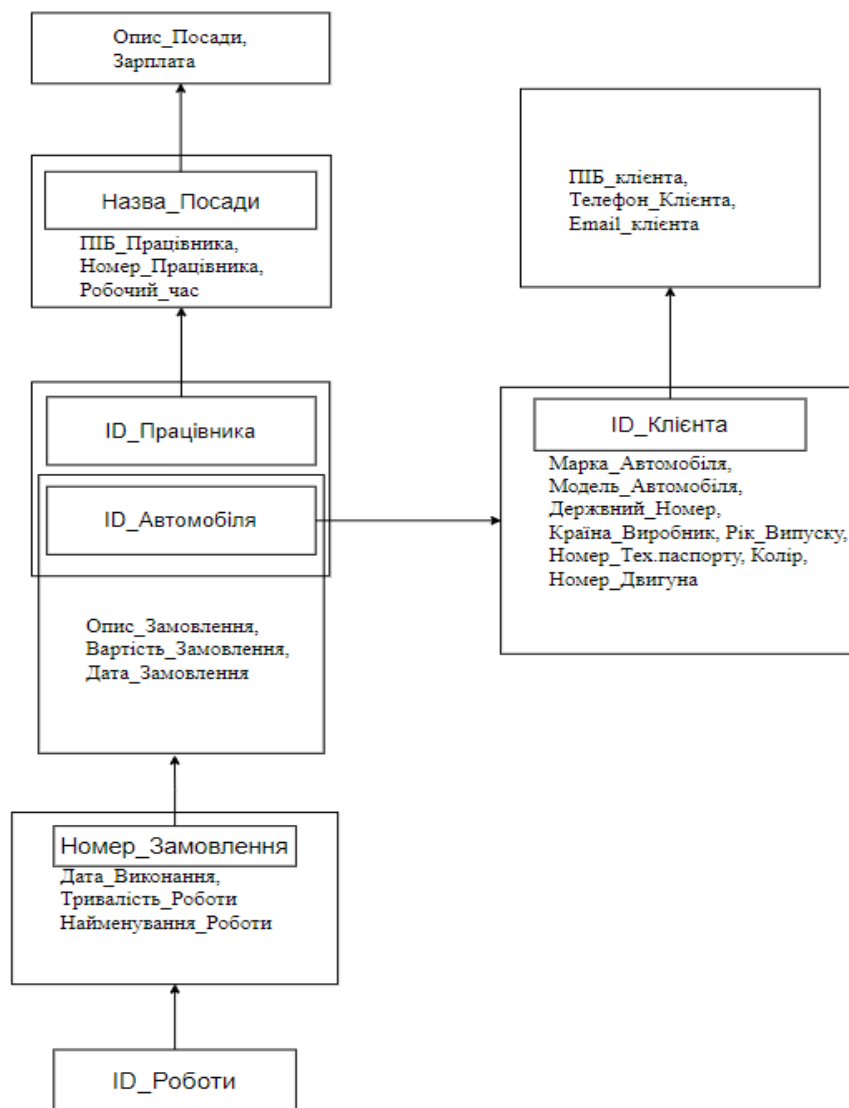


Рисунок 1.13 – Діаграма функціональних залежностей предметної області «База даних станції технічного обслуговування»

Відношення, які характеризують предметну область «База даних станції технічного обслуговування»:

R1: (<Назва_Посади>, Опис_Посади, Зарплата);

R2: (<ID_Працівника>, ПИБ_Працівника, Номер_Працівника, Робочий_час, Назва_Посади);

R3: (<ID_Автомобіля>, Марка_Автомобіля, Модель_Автомобіля, Держвний_Номер, Країна_Виробник, Рік_Випуску, Номер_Тех.паспорта, Колір, Номер_Двигуна, ID_Клієнта);

R4: (<ID_Клієнта>, ПИБ_клієнта, Телефон_Клієнта, Email_клієнта);

R5: (<Номер_Замовлення>, Опис_Замовлення, Вартість_Замовлення, Дата_Замовлення, ID_Автомобіля);

R6: (<ID_Роботи>, Дата_Виконання, Тривалість_Роботи, Найменування_Роботи, Номер_Замовлення);
 R7: (<ID_Працівника, ID_Автомобіля>).

1.37 Використовуючи діаграму функціональних залежностей (рис. 1.14), записати відношення, які характеризують предметну область «База даних футбольного клубу»

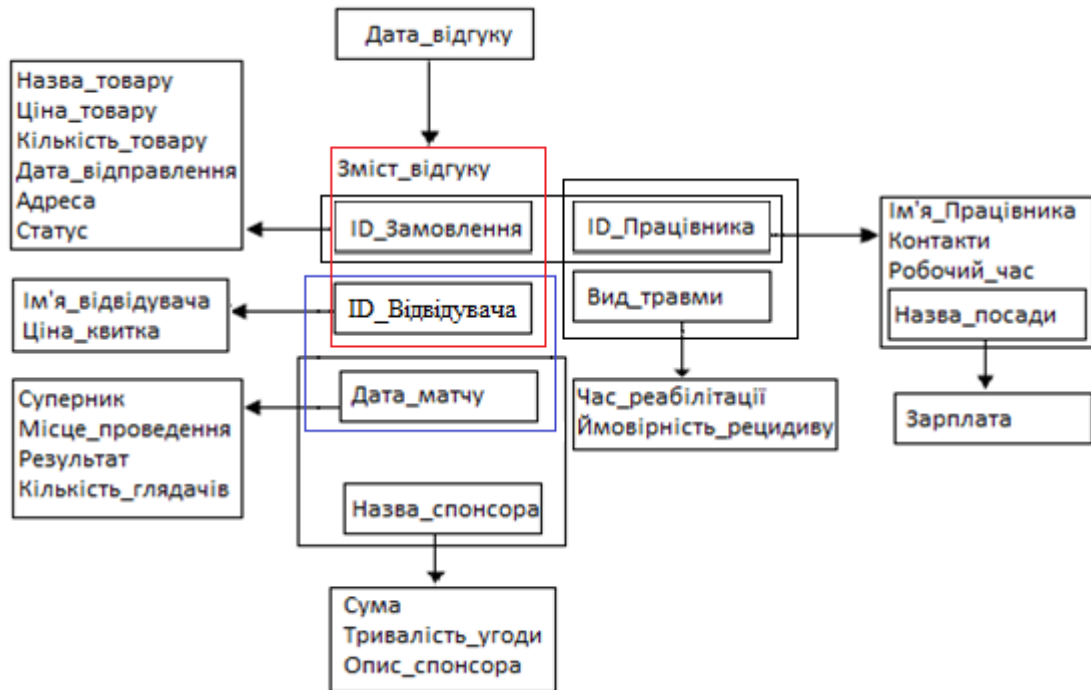


Рисунок 1.14 – Діаграма функціональних залежностей предметної області «База даних футбольного клубу»

Відношення, які характеризують предметну область «База даних футбольного клубу»:

R1: (<ID_працівника>, Ім'я_Працівника, Контакти, Робочий_час, Назва_посади);

R2: (<ID_Працівника, ID_Замовлення>);

R3: (<ID_Працівника, Вид_травми>);

R4: (<Назва_посади>, Зарплата);

R5: (<Дата_матчу>, Суперник, Місце_проведення, Результат, Кількість_глядачів);

R6: (<Дата_матчу, Назва_спонсора>);

R7: (<ID_Відвідувача, Дата_матчу>);

R8: (<ID_Замовлення>, Назва_товару, Ціна_товару, Кількість_товару, Дата_відправлення, Адреса, Статус);

R9: (<Дата_відгуку>, Зміст_відгуку, ID_Замовлення, ID_Відвідувача);

R10: (<Вид_травми>, Час_реабілітації, Ймовірність_рецидиву);

R11: (<ID_Відвідувача>, Ім'я_відвідувача, Ціна_квитка);

R12: (<Назва_спонсора>, Сума, Тривалість_угоди, Опис_спонсора).

1.38 За вказаними відношеннями, що задають предметну область бази даних магазину фруктів, побудувати діаграму функціональних залежностей:

- R1: (<Назва_фірми>, Телефон, Завершення_договору);
- R2: (<Назва_фірми, Код_товару>);
- R3: (<Код_товару>, Назва, Ціна);
- R4: (<Номер_поставки>, Маса, Дата_поставки);
- R5: (<Номер_поставки, Дата_продажу>);
- R6: (<Дата_продажу>, Кількість, Код_товару, ID_працівника);
- R7: (<ID_працівника>, ПІБ_працівника, Електронна_пошта, Робочий_час, Стаж, Назва_посади);
- R8: (<Назва_посади>, Зарплата).

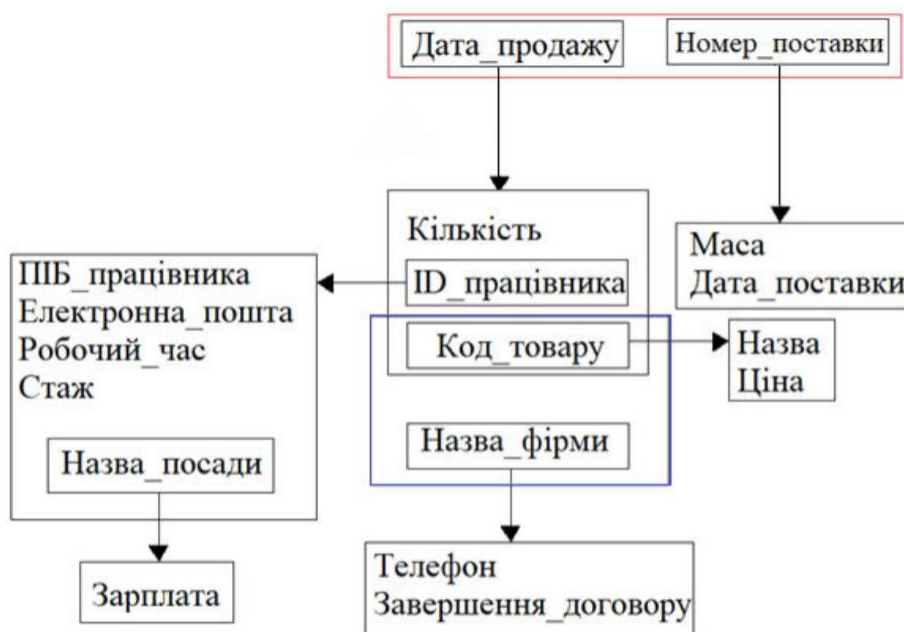


Рисунок 1.15 – Діаграма функціональних залежностей предметної області «База даних магазину фруктів»

1.39 Побудувати діаграму ER-типів предметної області «База даних магазину морепродуктів», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Чек (<ID_чек>, Дата_покупки, ПІБ_покупця);
2. Покупка (<ID_покупки>, Кількість);
3. Товар (<ID_товару>, Назва, Ціна);
4. Постачальник (<ID_пост>, Назва_пост, Адреса, Тел_пост);
5. Працівник (<Н_пасп_працівника>, ПІБ_працівника, Посада, Оклад, Тел_працівника).

Діаграма ER-типів предметної області «База даних магазину морепродуктів» наведена на рисунку 1.16.

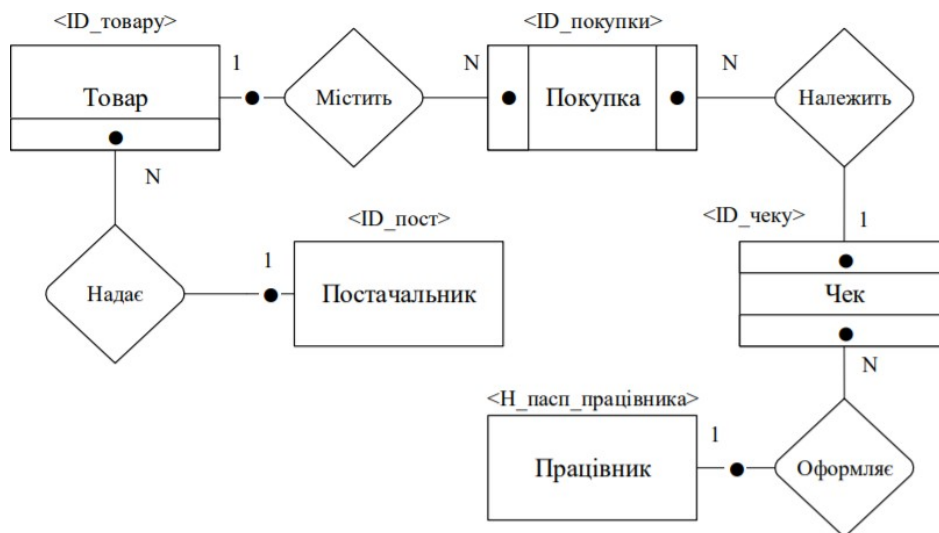


Рисунок 1.16 – Діаграма ER-типів предметної області «База даних магазину морепродуктів»

1.40 Використовуючи діаграму функціональних залежностей (рис. 1.17), записати відношення, які характеризують предметну область бази даних магазину морепродуктів

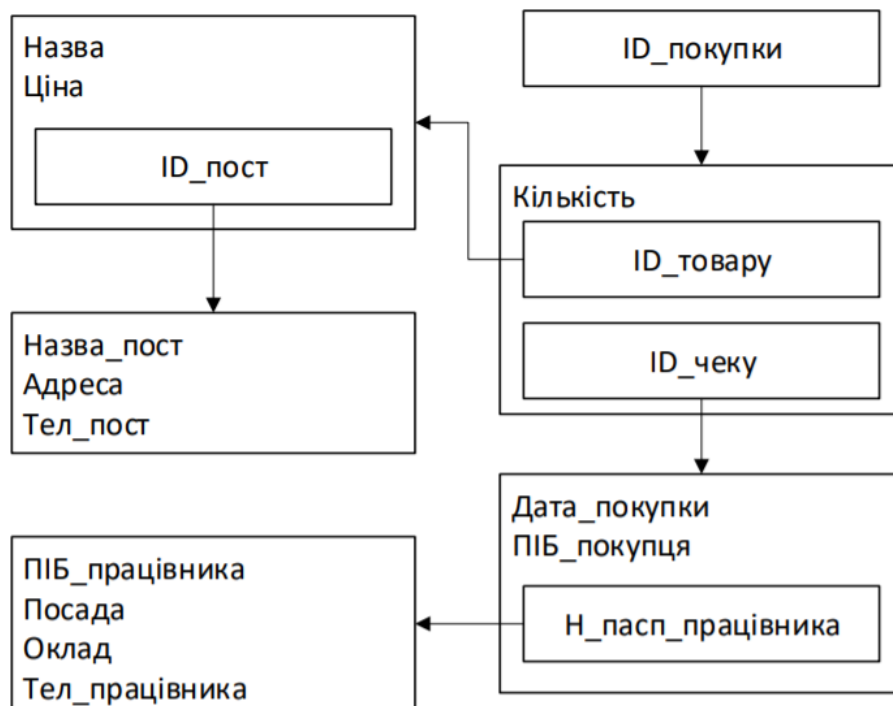


Рисунок 1.17 – Діаграма функціональних залежностей предметної області бази даних магазину морепродуктів

Відношення, які характеризують предметну область бази даних магазину морепродуктів:

R1 ($\langle ID_пост \rangle$, Назва_пост, Адреса, Тел_пост);

R2 ($\langle ID_товару \rangle$, Назва, Ціна, $ID_пост$);

R3 ($\langle ID_покупки \rangle$, Кількість, $ID_товару$, $ID_чеку$);

R4(<Н_пасп_працівника>, ПІБ_працівника, Посада, Оклад, Тел_працівника);
 R5 (<ID_чеку>, Дата_покупки, ПІБ_покупця, Н_пасп_працівника).

1.41 Побудувати діаграму ER-типів предметної області «База даних автомобільної заправки», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Товари (<ID_Товару>, Назва_товару, Кількість, Ціна);
2. Працівник (<ID_Працівника>, Ім'я_працівника, Контакти, Робочий_час, Стаж);
3. Посада (<Назва_посади>, Зарплата);
4. Клієнт (<ID_Клієнта>, Ім'я_клієнта, Телефон);
5. Знижка (<Відсоток_знижки>, Сума_для_знижки, Стаж_для_знижки).

Діаграма ER-типів предметної області «База даних автомобільної заправки» наведена на рисунку 1.18.



Рисунок 1.18 – Діаграма ER-типів предметної області «База даних автомобільної заправки»

1.42 За вказаними відношеннями, що задають предметну область бази даних автомобільної заправки, побудувати діаграму функціональних залежностей:

- R1: (<ID_Товару>, Назва_товару, Кількість, Ціна);
 R2: (<ID_Працівника>, <ID_Товару>);
 R3: (<ID_Працівника>, Ім'я_працівника, Контакти, Робочий_час, Стаж, Відсоток_знижки, Назва_посади);
 R4 (<Назва_посади>, Зарплата);
 R5 (<ID_Клієнта>, Ім'я_клієнта, Телефон);
 R6 (<ID_Клієнта, Відсоток_знижки>);
 R7 (<Відсоток_знижки>, Сума_для_знижки, Стаж_для_знижки).

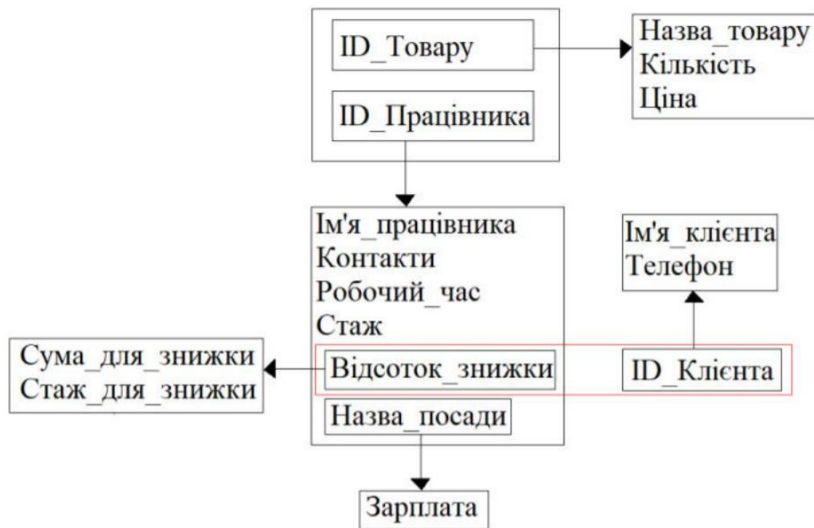


Рисунок 1.19 – Діаграма функціональних залежностей предметної області «База даних автомобільної заправки»

1.43 Побудувати ER-діаграму екземплярів для сутей Поле та Культура, які мають зв'язок під назвою «Потребує». Зв'язок між сутностями визначається як багато до одного, тому що одну культуру можна посіяти на кількох полях. Клас належності є обов'язковим тільки для поля, адже потрібно, щоб усі поля були засіяні для отримання урожаю, але використання усіх наявних видів культури не є обов'язковим.



Рисунок 1.20 – ER-діаграма екземплярів для сутей Поле та Культура

1.44 Побудувати діаграму ER-типів предметної області «База даних ательє з пошиття одягу», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Клієнт (<ПІБ_клієнта>, Адреса клієнта, Мобільний номер клієнта, Побаження);
2. Замовлення (<ID_замовлення>, Дата замовлення, Ціна, Строк виконання замовлення);
3. Закрійник-Майстер (<ПІБ_майстра>, Часна виконання Заміри, Мобільний номер майстра);
4. Швея (<ПІБ_швеї>, Строк здачі пошиття, Завантаженість швеї, Мобільний номер швеї);
5. Оплата (<ID_оплати>, Чек, Підпис клієнта, Видача замовлення).

Діаграма ER-типів предметної області «База даних ательє з пошиття одягу» наведена на рисунку 1.21

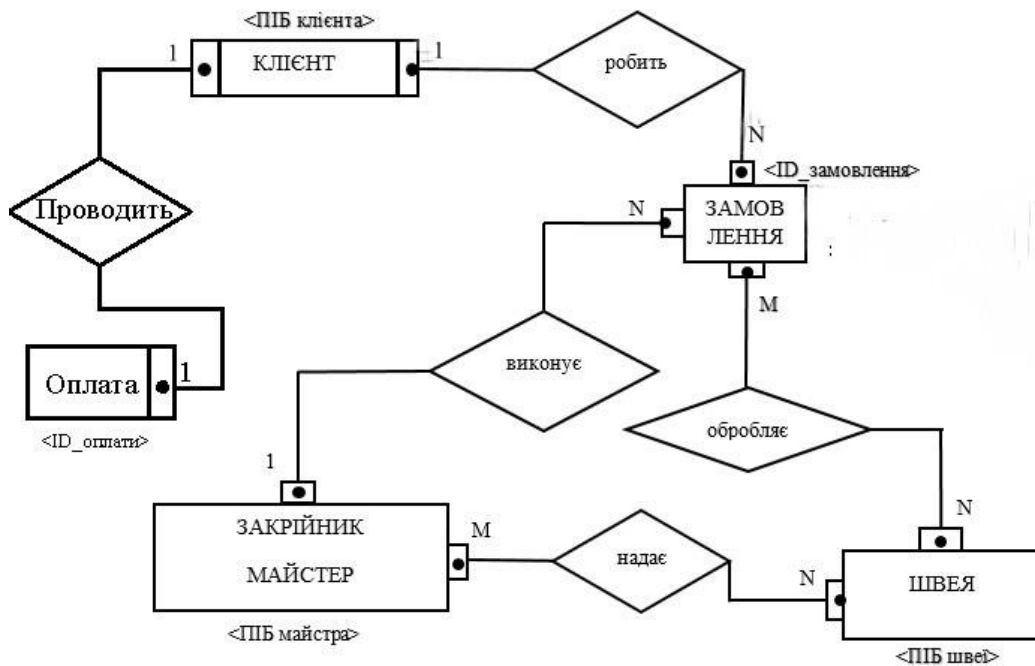


Рисунок 1.21 – Діаграма ER-типів предметної області «База даних ательє з пошиття одягу»

1.45 Використовуючи діаграму функціональних залежностей (рис. 1.22), записати відношення, які характеризують предметну область «База даних ательє з пошиття одягу»

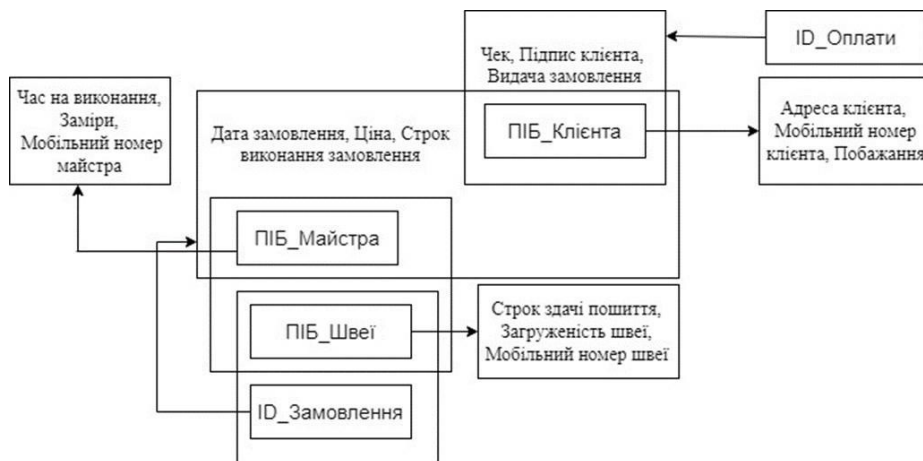


Рисунок 1.22 – Діаграма функціональних залежностей предметної області «База даних ательє з пошиття одягу»

Відношення, які характеризують предметну область «База даних ательє з пошиття одягу»:

R1(<ПІБ_клієнта>, Адреса клієнта, Мобільний номер клієнта, Побаження);

R2 (<ID_оплати>, Чек, Підпис клієнта, Видача замовлення, ПІБ_клієнта);

R3(<ПІБ_майстра>, Час на виконання, Заміри, Мобільний номер майстра);

R4(<ID_замовлення>, Дата замовлення, Ціна, Строк виконання замовлення, ПІБ_майстра, ПІБ_клієнта);

R5 (<ПІБ_швеї>, Строк здачі пошиття, Завантаженість швеї, Мобільний номер швеї);

R6 (<ПІБ_швеї>, <ID_замовлення>);

R7 (<ПІБ_майстра>, <ПІБ_швеї>).

1.46. Побудувати діаграму ER-типів предметної області «База даних букмекерської контори», використовуючи об'єкти, задані такими атрибутами предметної області:

1. Подія (<Код_Події>, Вид_спорту, Команда_1, Команда_2, Дата_події);
2. Результату (<ID_Результату>, Перемога_команди, Програш_команди, Коментар);
3. Виграш (<Код_Виграшу>, Сума_виграшу, Переможець, Підтвердження);
4. Ставка (<ID_Ставки>, Коефіцієнт, Сума_клієнта);
5. Букмекер (<ID_Букмекера>, ПІБ_Букмекера, Прогноз_на_подію, Успішність);
6. Клієнт (<ID_Клієнта>, ПІБ_Клієнта, Вік, Баланс, Номер, E-mail).

Діаграма ER-типів предметної області «База даних букмекерської контори» наведена на рисунку 1.23.

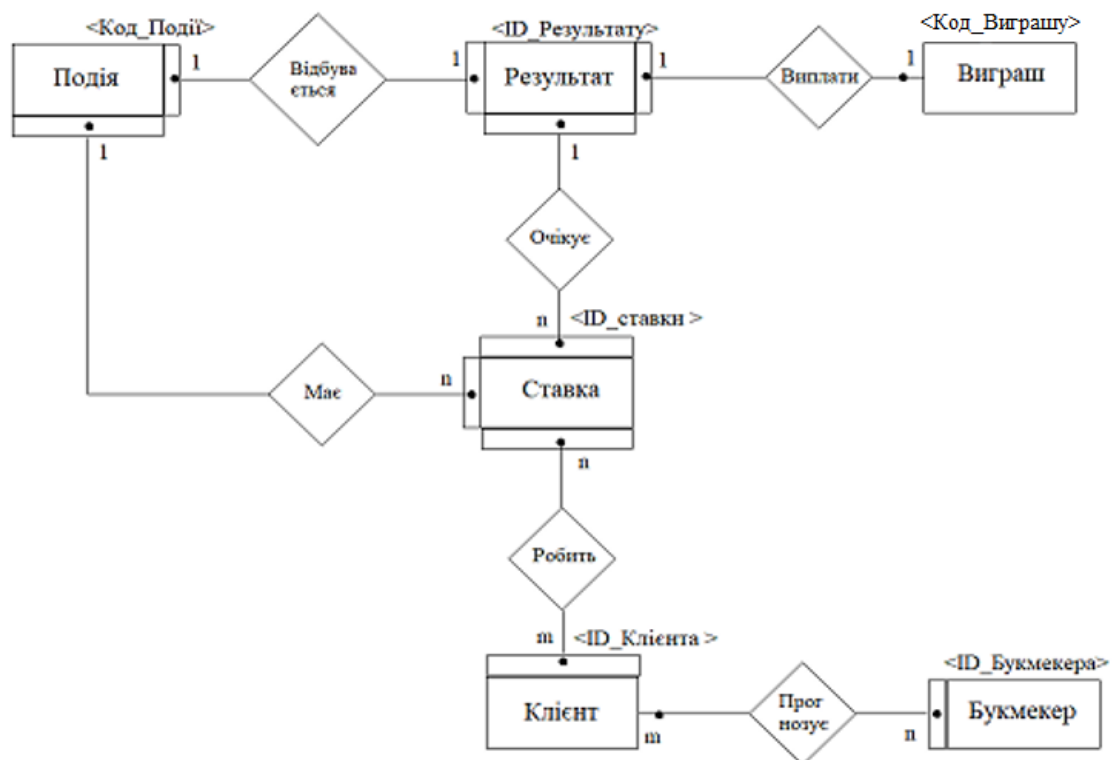


Рисунок 1.23 – Діаграма ER-типів предметної області «База даних букмекерської контори»

1.47 Побудувати ER-діаграму екземплярів для сутей Букмекер і Клієнт, якщо клас належності суті Букмекер є обов'язковим, а суті Клієнт – не обов'язковим. Тип зв'язку – n:m.

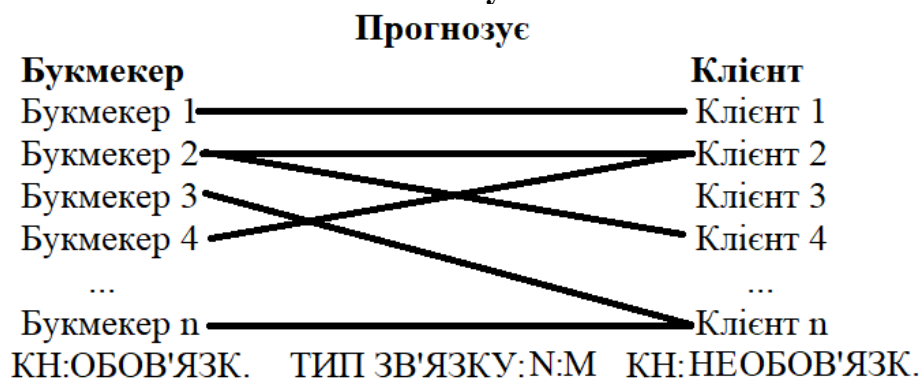


Рисунок 1.24 – Діаграма ER-екземплярів сутей Букмекер і Клієнт

1.48 Використовуючи діаграму функціональних залежностей, записати відношення, які характеризують предметну область «База даних букмекерської контори»



Рисунок 1.25 – Діаграма функціональної залежності предметної області «База даних букмекерської контори»

Відношення, що задають предметну область «База даних букмекерської контори»:

- R1 (<Код_Події>, Вид_спорту, Команда_1, Команда_2, Дата_події);
- R2 (<<ID_Результату>, Перемога_команди, Програш_команди, Коментар, Код_Події, ID_Ставки, Код_Виграшу);
- R3 (<Код_Виграшу>, Сума_виграшу, Переможець, Підтвердження);
- R4 (<ID_Ставки>, Коефіцієнт, Сума_клієнта, Код_Події);
- R5 (<ID_Букмекера>, ПІБ_Букмекера, Прогноз_на_подію, Успішність);
- R6 (<ID_Клієнта>, ПІБ_Клієнта, Вік, Баланс, Номер, E-mail).

2 МОВА ЗАПИТІВ SQL

2.1 Створити таблицю Студенти зі стовпцями Номер_залікової_книжки, ПІБ_студента та Група

Створення таблиць мовою SQL здійснюється за допомогою операції CREATE TABLE, яка має такий синтаксис:

```
CREATE TABLE <ім'я таблиці>  
(<ім'я стовпця><тип даних>[NOT NULL]  
[,<ім'я стовпця><тип даних>[NOT NULL]]...);
```

Обов'язковими операндами оператора є ім'я створюваної таблиці та ім'я хоча б одного стовпця з вказанням типу даних, що зберігатимуться у ньому. Як правило, при створенні таблиці для стовпців можуть вказуватись деякі додаткові правила контролю введення в них даних, наприклад конструкція NOT NULL (не пуста), яка вимагатиме, щоб у стовпці обов'язково було визначене значення.

Код такого запиту, створений у редакторі Microsoft Access, матиме вигляд:

```
CREATE TABLE Студенти (  
    Номер_залікової_книжки varchar (10) NOT NULL,  
    ПІБ_студента varchar (50) NOT NULL,  
    Група varchar (10) NOT NULL );
```

2.2 Додати до існуючої таблиці Студенти стовець Адреса

Додавання стовпців до раніше створеної таблиці відбувається командою:

```
ALTER TABLE < ім'я таблиці >  
    (ADD <ім'я стовпця> [<тип даних>][NOT NULL]).
```

```
ALTER TABLE Студенти ADD Адреса varchar (50) NOT NULL;
```

2.3 Видалити таблицю Студенти повністю

Оператор видалення таблиці має вигляд: DROP TABLE <ім'я таблиці>.

```
DROP TABLE Студент;
```


2.4 Додати в таблицю Студенти (Номер_залікової_книжки, ПІБ_студента, Група) запис про студента Паламарчука Сергія Івановича, що навчається у групі 2ПІ-21б з номером залікової книжки 112-680

Для додавання нових записів в таблицю використовується оператор INSERT INTO, який має такий формат:

```
INSERT INTO <ім'я таблиці>  
[(<список стовпців>)]  
VALUES (<список значень>)
```

У першому форматі оператор INSERT призначений для введення нових записів з заданими значеннями у стовпцях. Порядок перерахування імен стовпців має відповідати порядку значень, перерахованих в списку операнда VALUES. Якщо <список стовпців> упущений, то у <списку значень> мають бути перераховані всі значення в порядку стовпців структури таблиці.

```
INSERT INTO Студенти (Номер_залікової_книжки, ПІБ_студента,  
Група)  
VALUES ('112-680', 'Паламарчук Сергій Іванович', '2ПІ-21б');
```

або

```
INSERT INTO Студенти  
VALUES ('112-680', 'Паламарчук Сергій Іванович', '2ПІ-21б');
```

Також можна вносити неповні дані у рядок, тобто деякі стовпці можуть залишатись порожніми. Тоді, обов'язково, у запиті має бути присутній список стовпців, які підлягатимуть заповненню, і список значень має йти у тій же послідовності, як і перераховані стовпці у списку значень.

Але це можливо за умови, що при створенні таблиці допускались порожні значення у стовпцях. Якщо ж стовпець був визначений з додатковим параметром NOT NULL, то такий стовпець обов'язково має бути заповнений і якщо в запиті на внесення даних він буде пропущений, то запит не буде виконано.

2.5 Видалити з таблиці Студенти (Номер_залікової_книжки, ПІБ_студента, Група) запис про студента з прізвищем Паламарчук Сергій Іванович

Видалення записів з таблиці можна здійснити за допомогою оператора DELETE, який має формат:

```
DELETE FROM <ім'я таблиці>  
[WHERE <умова>];
```

Результатом виконання оператора DELETE є видалення з вказаної таблиці рядків, які задовольняють умову, визначену операндом WHERE. Якщо операнд WHERE опущений, видаленню підлягають всі записи таблиці.

У даному випадку необхідно здійснити пошук визначеного рядка, тому потрібно застосувати оператор WHERE з перевіркою на збіг ПІБ_студента = 'Паламарчук Сергій Іванович'.

```
DELETE FROM Студенти
WHERE ПІБ_студента = 'Паламарчук Сергій Іванович';
```

2.6 Студента Мельничука Артема Олеговича перевели з групи 2ПІ-216 до групи 1ПІ-216, у зв'язку з чим потрібно внести відповідні зміни у таблицю Студенти (Номер_залікової_книжки, ПІБ_студента, Група).

Для внесення змін у наявні записи використовується оператор UPDATE, який має формат виду:

```
UPDATE <ім'я таблиці>
SET <ім'я стовпця> = {<вираз>, NULL}
[, SET <ім'я стовпця> = {<вираз>, NULL}...]
[WHERE <умова>];
```

Виконання оператора UPDATE полягає у зміні значення у визначених оператором SET стовпцях таблиці для тих записів, які задовольняють умову, задану оператором WHERE. Нові значення полів в записах можуть бути порожніми (NULL), або обраховуватись відповідно до арифметичного виразу.

Для заданого прикладу потрібно знайти рядок, для якого збігаються ПІБ_студента та Група, і встановити туди нове значення групи.

```
UPDATE Студенти
SET Група = '1ПІ-216'
WHERE ПІБ_студента='Мельничук Артем Олегович' AND
Група='2ПІ-216';
```

2.7 Дано таблицю Студенти (Номер_залікової_книжки, Прізвище, Ім'я, По батькові, Група). Вивести інформацію про всіх студентів двома способами

Щоб вивести бажану інформацію про студентів, потрібно у списку оператора SELECT перерахувати назви усіх бажаних стовпців таблиці. Оскільки в умові задачі чітко не зазначено, яку саме інформацію про студентів потрібно виводити, то у списку оператора SELECT мають бути

перераховані усі стовпці таблиці Студенти. Код такого запиту, створений у редакторі Microsoft Access, матиме вигляд

```
SELECT [Номер залікової книжки], Прізвище, [Ім'я], [По  
батькові],  
Група  
FROM Студенти;
```

Варто звернути увагу, що у випадку складної назви стовпця таблиці (назва складається з декількох слів чи наявні орфографічні знаки), в редакторі Microsoft Access така назва стовпця може братись у квадратні дужки.

Мова запитів SQL дозволяє замінити явний перерахунок усіх стовпців таблиці спеціальним символом «*», що значно спрощує форму запису запиту. Код такого запиту матиме вигляд:

```
SELECT *  
FROM Студенти;
```

Використання спеціального символу «*» є дуже зручним способом при конструюванні та/або відлагодженні запиту. Однак при створенні запитів у додатках краще явно перераховувати усі стовпці, щоб іншим користувачам було зрозуміло, яку саме інформацію буде виведено на екран, оскільки не завжди структура таблиці є відомою. Попри простоту запиту з використанням спеціального символу «*», продуктивність запиту при цьому не збільшується порівняно з явним перерахунком усіх стовпців.

2.8 Дано таблицю Студенти (Номер_залікової_книжки, Прізвище, Ім'я, По батькові, Група). Вивести назви груп

В таблиці Студенти одна й та ж група може повторюватись багато разів, тому з результату запиту потрібно вилучити повторювані значення. Це можна зробити з використанням ключового слова DISTINCT перед назвою стовпця Група. Код такого запиту матиме вигляд

```
SELECT DISTINCT Група  
FROM Студенти;
```

2.9 Дано таблицю Студенти (Номер_залікової_книжки, Прізвище, Ім'я, По батькові, Група). Вивести інформацію про всіх студентів з прізвищем Мельник трьома способами

Перш за все потрібно зрозуміти, які стовпці потрібно вносити до результату запиту. Оскільки чіткі вказівки в умові задачі відсутні, то виводити потрібно усі стовпці або явним перерахунком усіх стовпців, або з використанням спеціального символу «*».

Оскільки маємо умову відбору рядків за прізвищем, то потрібно застосувати конструкції WHERE. Найпростіший варіант здійснити відбір рядків передбачає використання оператора «=»

```
SELECT [Номер залікової книжки], Прізвище, [Ім'я], [По  
батькові],  
Група  
FROM Студенти  
WHERE Прізвище = «Мельник»;
```

Також можна виконати цей запит з використанням шаблону LIKE

```
SELECT [Номер залікової книжки], Прізвище, [Ім'я], [По  
батькові],  
Група  
FROM Студенти  
WHERE Прізвище LIKE «Мельник»;
```

Результат запиту буде ідентичний з попереднім варіантом, однак записувати його довше. Шаблон LIKE прийнято застосовувати у випадках, коли умова для відбору відома не повністю.

Ще один варіант запису запиту на відбір рядків можливий з використанням конструкції IN

```
SELECT [Номер залікової книжки], Прізвище, [Ім'я], [По  
батькові],  
Група  
FROM Студенти  
WHERE Прізвище IN («Мельник»);
```

Конструкція IN визначає належність до деякої множини, тому її доцільніше використовувати за наявності двох і більше умов, наприклад, у випадку виведення на екран інформації про всіх студентів з прізвищами Іванов, Мельник та Петренко. Тоді останній запит мав би вигляд

```
SELECT [Номер залікової книжки], Прізвище, [Ім'я], [По  
батькові],  
Група  
FROM Студенти  
WHERE Прізвище IN («Іванов», «Мельник», «Петренко»);
```

2.10 Дано таблицю Студенти (Номер_залікової_книжки, Прізвище, Ім'я, По батькові, Група). Вивести інформацію про всіх студентів, чие прізвище починається на літеру «М» та має довжину не менше 5 символів

У даній задачі умова для відбору відома частково, тому потрібно застосувати шаблон LIKE. В сучасних версіях MS Access в шаблоні LIKE

можуть застосовуватись такі підстановкові спеціальні символи, як «*» та «?», хоча в більш ранніх версіях цього редактора та в інших, наприклад PostgreSQL, застосовуються спеціальні символи «%» та «_» відповідно, що виконують аналогічні функції. Спеціальний символ «*» замінює собою будь-яку кількість символів від нуля і більше, а спеціальний символ «?» позначає лише один будь-який символ. Код запити для задачі матиме вигляд

```
SELECT *
FROM Студенти
WHERE Прізвище LIKE "M????*";
```

2.11 Дано таблицю Студенти (Номер_залікової_книжки, Прізвище, Ім'я, По батькові, Група). Стало відомо, що при внесенні інформації до таблиці до прізвищ деяких студентів було помилково додано символ «?», а тому потрібно знайти ці рядки, щоб у подальшому виправити помилки

Трапляється так, що в шуканому шаблоні присутній символ «*» або «?», які є частиною шаблону, і з них потрібно зняти їх спеціальне призначення. Для цього в шаблоні конструкції LIKE відповідний спеціальний символ береться у квадратні дужки.

Оскільки шукані рядки мають однозначно ідентифікуватись навіть за умови наявності декількох студентів з однаковими прізвищами, доцільно до списку SELECT додати стовпець Номер_залікової_книжки, значення якого є унікальним для кожного студента. В такому випадку можна застосувати такий запит

```
SELECT [Номер_залікової_книжки], Прізвище
FROM Студенти
WHERE Прізвище LIKE "[?]*";
```

У прикладі спеціальний символ «?», взятий у квадратні дужки, втратив своє спеціальне призначення і сприймається як звичайний шаблон для відбору. Не важливо, де саме в прізвищі стоїть символ «?», тому всі символи до та після символу «?» позначені підстановковим спеціальним символом «*».

В шаблоні LIKE також можна закодувати одну цифру за допомогою спеціального символу «#», діапазон літер – [a–я], за межами діапазону літер – [!a–я], не цифру – [!0–9].

2.12 Дано таблицю Співробітник (Табельний_номер, ПІБ, Відділ, Посада, Дата_прийняття, Зарплата, Догана). Вивести інформацію про співробітників Технічного відділу, які отримують зарплату від 8500,00 до 12000,00 гривень, мають догану і не працюють на посаді директора відділу або старшого технолога

Оскільки в умові задачі присутні декілька умов, то їх поєднання відбувається з використанням умовних операторів AND або OR.

Для відбору працівників Технічного відділу використовуємо оператор рівності «=». Для відбору працівників із визначеною зарплатою скористаємось оператором BETWEEN ... AND. Перевірка на наявність доган може здійснюватись конструкцією IS NOT NULL, а неналежність до множини зазначених посад – оператором NOT IN.

```
SELECT *
FROM Співробітники
WHERE Відділ = "Технічний" AND
      Зарплата BETWEEN 8500 AND 12000 AND
      Догана IS NOT NULL AND
      Посада NOT IN ("Директор", "Старший технолог");
```

2.13 Дано Таблицю Дитина (Табельний_номер, ПІБ, Група, Дата_народження). Вивести всю інформацію про найстаршу дитину

Для виконання цього запиту потрібно відсортувати інформацію про всіх дітей за зменшенням дати народження за допомогою оператора ORDER BY і ключового слова DESC, а потім обмежити виведення відомостей лише першим рядком, для чого потрібно застосувати оператор TOP N, де N – кількість перших рядків у відсортованому списку, які підлягають виведенню на екран. У нашому випадку це TOP 1.

Оскільки в умові сказано, що вивести потрібно всю інформацію про дитину, то після оператора TOP 1 потрібно використати спеціальний символ «*». В іншому випадку потрібно було б чітко перераховувати назви стовпців для виведення.

```
SELECT TOP 1 *
FROM Дитина
ORDER BY (Дата_народження_дитини) DESC;
```

2.14 Дано Таблицю Дитина (Табельний_номер, ПІБ, Група, Дата_народження). Вивести три найбільші групи в порядку спадання кількості дітей у групах

Цей запит дещо схожий на попередній, адже також потрібно обмежити виведення результатів лише трьома рядками з інформацією про найбільші групи, тому теж застосуємо оператор TOP 3. Однак інформація про

кількість дітей у групах відсутня і її потрібно порахувати для кожної групи.

Підрахунок кількості дітей у групі можна виконати з використанням агрегатної функції COUNT (Група) в поєднанні з оператором GROUP BY, який групуватиме ці підрахунки саме для кожної групи.

Стовпець COUNT (Група) є обчислюваним і йому потрібно явно задати назву [Кількість дітей у групі] за допомогою конструкції AS.

Сортування результатів відбувається за обчислюваним стовпцем COUNT (Група) з ключовим словом DESC.

```
SELECT TOP 3 Група, COUNT (Група) AS [Кількість дітей у групі]
FROM Дитина
GROUP BY Група
ORDER BY COUNT (Група) DESC;
```

Знайти найбільші заробітні плати за віковими групами з таблиці Співробітники (Заробітна_плата, Вік, ...).

```
SELECT Вік, MIN(Заробітна_плата), MAX(Заробітна_плата)
FROM Співробітники
GROUP BY Вік;
```

2.15 При вибірці з таблиці Співробітники (Заробітна_плата, Вік, ...) створити нове поле Максимальна_зарплата, в яке буде поміщено значення максимальної заробітної плати для віку 30 років

```
SELECT MAX(Заробітна_плата) AS Максимальна_зарплата
FROM Співробітники
WHERE Вік = 30;
```

2.16 Дано таблицю Кабінет (Номер, Кількість_місць, Поверх). Вивести інформацію про кабінети, які більші за середні

Відбір кабінетів здійснюється за допомогою оператора WHERE, але для цього потрібно знати середнє значення кількості місць у кабінетах. Середнє значення кількості місць легко можна знайти, використавши агрегатну функцію AVG (Кількість_місць). Однак безпосередньо її використати при побудові умови порівняння неможливо. Для цього її потрібно помістити у вкладений оператор SELECT.

```
SELECT *
FROM Кабінет
WHERE [Кількість_місць] > (SELECT AVG(Кількість_місць)
FROM Кабінет);
```

2.17 Дано таблицю Фауна угідь (Назва_виду, Назва_угіддя, Популяція). Вивести інформацію про види тварин, популяція яких на всіх угіддях перевищує 1000 особин, в порядку зростання популяції

Враховуючи, що один і той же вид тварин може мешкати у різних угіддях, загальну популяцію для кожного виду потрібно розраховувати шляхом застосування агрегатної функції SUM (Популяція) в поєднанні з оператором GROUP BY [Назва виду], який групуватиме інформацію про загальну популяцію для кожного виду. Щоб було зрозуміло, що за значення було розраховане, обчислюваному стовпцю SUM (Популяція) явно задамо назву за допомогою конструкції AS [Загальна чисельність популяції].

В подальшому потрібно вилучити зі списку для виведення ті групи, де розрахована загальна популяція (для кожному виду тварин) менша або дорівнює 1000 особин. Це можна зробити за допомогою оператора HAVING SUM (Популяція) >1000.

За умовою результати потрібно вивести в порядку зростання популяції, тому застосуємо оператор ORDER BY SUM (Популяція). Потрібно звернути увагу, що в операторі ORDER BY відсутнє ключове слово, яке задає порядок сортування за зростанням, оскільки сортування за зростанням здійснюється за замовчуванням. Хоча можна й явно вказувати таке сортування з використанням ключового слова ASC.

```
SELECT [Назва_виду], SUM (Популяція) AS [Загальна  
чисельність популяції]  
FROM [Фауна угідь]  
GROUP BY [Назва_виду]  
HAVING SUM (Популяція) >1000  
ORDER BY SUM (Популяція);
```

2.18 Дано таблицю Мисливські угіддя (Код угіддя, Назва, Місце знаходження, Площа). Створити запит «Статистика угідь», який підраховує загальну кількість угідь Вінницької та Миколаївської областей, загальну їх площу, а також їх частку серед усіх угідь

Для підрахунку кількості угідь потрібно застосувати агрегатну функцію COUNT (*), а для підрахунку площі угідь – SUM (Площа). Для функції COUNT не важливо, за яким стовпцем будемо робити підрахунок, адже рахується кількість записів, що відповідають вимозі, тому в дужках як аргумент цієї функції вказується «*». Хоча замість символу «*» може вказуватись будь-який інший стовпець, що приведе до отримання ідентичного результату. Для підрахунку площі важливо, щоб аргументом функції був саме стовпець з відповідними даними, тобто Площа.

Якщо в результаті виконання запиту створюються обчислювані стовпці, то прийнято їм явно задавати назву. Для цього використовується конструкція AS.

Оскільки розрахунки проводяться лише для угідь Вінницької та Миколаївської областей, то для відбору відповідних угідь потрібно використати конструкцію WHERE, в якій за допомогою конструкції IN можна відібрати угіддя з заданих областей.

Найбільшу складність викликає розрахунок частки угідь Вінницької та Миколаївської областей, адже потрібно знайти загальну площу угідь двох областей і загальну площу всіх областей. Для розрахунку загальної площі всіх областей потрібно використати вкладений запит. Щоб округлити отримане значення частки у відсотках до двох знаків після коми, застосуємо функцію ROUND, перший аргумент якої – власне розраховане значення частки у відсотках, а другий – кількість знаків після коми для округлення.

Потрібно звернути увагу, що у зовнішньому операторі SELECT усі розрахунки SUM (Площа) розраховуються лише для угідь Вінницької та Миколаївської областей, а ця ж операція у вкладеному операторі SELECT діє для всіх угідь.

```
SELECT COUNT (*) AS [Загальна кількість угідь],
       SUM (Площа) AS [Загальна площа угідь],
       ROUND (SUM (Площа) / (SELECT SUM (Площа)
                               FROM [Мисливські
угіддя])*100,2)
                               AS [Частка від всіх угідь]
FROM [Мисливські угіддя]
WHERE [Місце знаходження] IN ("Миколаївська область",
                               "Вінницька область");
```

2.19 Дано таблиці Читач (ID_читача, ПІБ_читача, ...) та Облік (ID_читача, Дата_видачі, Дата_повернення, Статус_замовлення, Шифр_видачі). Потрібно вивести ПІБ читача та всі його замовлення, що були опрацьовані за останні 30 днів. Опрацьованими вважаються замовлення, у яких дата видачі або дата повернення книги була здійснена за останні 30 днів

Вся інформація про замовлення міститься в таблиці Облік, однак за умовою нам потрібно вивести ще й ПІБ читача, який відсутній у цій таблиці. В таблиці Облік наявна лише інформація про ID_читача. Тому необхідно у запиті пов'язати дві таблиці Читач і Облік за допомогою їх спільного ключа ID_читача з використанням операції INNER JOIN.

Відбір опрацьованих замовлень за останні 30 днів можна виконати в конструкції WHERE за допомогою комбінації операторів BETWEEN ... AND в поєднанні з вбудованою функцією Date (), яка повертає поточну дату, та Date () – 30, яка визначає дату, що була 30 днів тому. Знайдені

значення дат і є аргументами в операторах BETWEEN ... AND, які застосовуються як до стовпця Дата_видачі, так і до стовпця Дата_повернення, але об'єднані логічним оператором OR.

Сортування результатів здійснимо за ПІБ читача в алфавітному порядку.

```
SELECT ПІБ_читача, Дата_видачі, Дата_повернення,  
        Статус_замовлення, Шифр_видачі  
FROM Читач INNER JOIN Облік ON Читач.ID_читача =  
        Облік.ID_читача  
WHERE (Дата_видачі) BETWEEN Date() AND Date()-30  
        OR (Дата_повернення) BETWEEN Date() AND Date()-30  
ORDER BY ПІБ_читача;
```

2.20 Дано таблицю Замовлення (ID_Замовлення, Категорія, Вага, Ціна_продажу, ID_Клієнта, Дата_Замовлення). Вивести стосовно клієнта інформацію про загальну ціну продажу всіх його замовлень, загальну вартість пакування, загальну вартість доставки та повну вартість з доставкою та пакуванням. ID_Клієнта вводиться користувачем. Пакування розраховується як 5% від ціни продажу, вартість доставки одного замовлення становить 75 грн

У кожного клієнта може бути декілька замовлень, тому, щоб знайти загальні вартості всіх замовлень, потрібно згрупувати результати за значенням ID_Клієнта в конструкції GROUP BY.

ID_Клієнта слугує параметром відбору рядків, до яких застосовуватимуться обчислення, тому можна використати оператор WHERE. Однак саме значення ID_Клієнта є невідомим, оскільки має бути введено користувачем в момент запуску запиту. Тому в запиті має бути реалізована можливість формування діалогового вікна, куди користувач зможе ввести ID_Клієнта, а саме: WHERE ID_Клієнта Like "*" & [Введіть ID_Клієнта] & "*"

З іншої сторони відібрати можна не самі рядки, а вже сформовані групи. Тоді можна застосувати конструкцію HAVING ID_Клієнта Like "*" & [Введіть ID_Клієнта] & "*".

Однак перший варіант є більш доцільний, оскільки програма спочатку відбере потрібні рядки і потім до них застосує обчислення. У випадку конструкції HAVING програма мусить групувати і виконувати обчислення для кожного клієнта, а потім вже відсіє більшість отриманих результатів, що вимагає значно більше обчислювальних ресурсів, хоча сформований результат буде ідентичним.

Для самих розрахунків потрібно використати агрегатну функцію SUM і звичайні арифметичні операції.

Щоб розуміти зміст отриманих чисел, потрібно явно задати назви обчислюваним полям за допомогою конструкції AS.

```

SELECT ID_Клієнта, SUM(Ціна_продажу) AS [Загальна ціна
продажу], SUM(Ціна_продажу)*0.05 AS [Загальна вартість
пакування], SUM(75) AS [Загальна вартість доставки], SUM
(Ціна_продажу*1.05+75) AS [Повна вартість з доставкою та
пакуванням]
FROM Замовлення
WHERE ID_Клієнта Like "*" & [Введіть ID_Клієнта] & "*"
GROUP BY ID_Клієнта;

```

або

```

SELECT ID_Клієнта, SUM(Ціна_продажу) AS [Загальна ціна
продажу], SUM(Ціна_продажу)*0.05 AS [Загальна вартість
пакування], SUM(75) AS [Загальна вартість доставки], SUM
(Ціна_продажу*1.05+75) AS [Повна вартість з доставкою та
пакуванням]
FROM Замовлення
GROUP BY ID_Клієнта
HAVING ID_Клієнта Like "*" & [Введіть ID_Клієнта] & "*";

```

2.21 Запропонувати варіант реалізації запиту «Програма знижок» для відношень Відвідувач (ID_Відвідувача, ПІБ_Відвідувача, ID_Оплати...) та Оплати (ID_Оплати, ID_Відвідувача, Сума_оплати, Відсоток_знижки)

Запит «Програма знижок» показує знижку на наступне замовлення. Припустимо, що знижку більшу за 5% можуть отримувати лише клієнти з картою накопичення, то за допомогою оператора BETWEEN потрібно відшукати таких клієнтів. Верхньою межею є 10%, адже це максимальна можлива знижка у готелі. Оскільки клієнт не може мати знижку більшу за 10%, тоді за допомогою оператора IIF додаємо знижку лише клієнтам, у яких відсоток знижки менший десяти. Якщо клієнт не має карти накопичення, знижку він отримує лише у випадку, якщо його оплата перевищує 700 гривень. Відсортуюмо результати за спаданням відсотка знижки.

```

SELECT ПІБ_Відвідувача, Сума_оплати, Відсоток_знижки,
ROUND(Сума_оплати-
((Сума_оплати*Відсоток_знижки)/100))
AS [Ціна із знижкою],
IIF(Відсоток_знижки<10,Відсоток_знижки+1,Відсоток_знижки)
AS [Знижка на наступне поселення]
FROM Оплата INNER JOIN Відвідувач ON Оплата.ID_Оплати =
Відвідувач.ID_Оплати
WHERE Відсоток_знижки BETWEEN 5 AND 10
OR Сума_оплати-
((Сума_оплати*Відсоток_знижки)/100)>700
ORDER BY Відсоток_знижки DESC;

```

2.22 Запропонувати варіант реалізації запиту «Найбільші скарги»

Одним з варіантів реалізації запиту «Найбільші скарги» є пошук скарг, які більші за розміром, ніж скарга, як була написана відвідувачем з ID_Відвідувача = 2. Порівняння змісту скарг відбувається за допомогою функції LEN, яка повертає кількість символів у тексті.

```
SELECT Готель.Назва_Готелю, Скарга.Зміст_скарги,  
                                           ПІБ_Відвідувача  
FROM Готель INNER JOIN ((Відвідувач INNER JOIN Скарга ON  
Відвідувач.ID_Відвідувача = Скарга.ID_Відвідувача) INNER JOIN  
[Відвідувач-Готель] ON Відвідувач.ID_Відвідувача = [Відвідувач-  
Готель].ID_Відвідувача) ON Готель.ID_Готелю = [Відвідувач-  
Готель].ID_Готелю  
WHERE LEN(Зміст_скарги) > ( SELECT LEN(Зміст_скарги)  
                               FROM Скарга  
                               WHERE ID_Відвідувача = 2);
```

2.23 Запропонувати варіант реалізації запиту «Місця в готелі»

Запит «Місця у готелі» виводить інформацію про всі готелі, кількість зайнятих та вільних місць, а також їх загальну кількість. Кількість відвідувачів рахується функцією COUNT, інші обрахунки відбуваються математичними обчисленнями.

```
SELECT Готель.Назва_готелю, COUNT(ID_Відвідувача) AS  
Кількість_відвідувачів, Готель.Кількість_вільних_місць,  
(Кількість_вільних_місць+COUNT(ID_Відвідувача)) AS  
Усі_місця_у_готелі  
FROM Готель INNER JOIN [Відвідувач-Готель] ON  
Готель.ID_Готелю  
= [Відвідувач-  
Готель].ID_Готелю  
GROUP BY Готель.Назва_готелю, Готель.Кількість_вільних_місць;
```

2.24 Запропонувати варіант реалізації запиту «Розрахунок премії для відповідальних працівників»

Реалізація такого запиту відбувається на основі даних таблиць Працівник і Робота.

Для всіх працівників, які відпрацювали більше 30 днів, може нараховуватись премія. Однак спочатку потрібно порахувати, за допомогою агрегатної функції Sum, скільки днів відпрацював працівник. За допомогою функції If задається алгоритм нарахування премії, а саме: для працівників, які відпрацювали більше 60 днів, премія становить 30% від базової заробітної плати; більше 50 і менше 60 днів – 20%; більше 30 і менше або дорівнює 50 – 10%.

```

SELECT Працівник.ПІБ_Працівника, Sum(Робота.Тривалість) AS
Робочих_Днів, Працівник.Заробітна_Плата,
IIf(Sum(Робота.Тривалість)>=60,Працівник.Заробітна_Плата*0.3,
IIf(Sum(Робота.Тривалість)>50 And
Sum(Робота.Тривалість)<60,Працівник.Заробітна_Плата*0.2,Пра
цівник.Заробітна_Плата*0.1)) AS Премія,
Працівник.Заробітна_Плата+IIf(Sum(Робота.Тривалість)>=60,Пр
ацівник.Заробітна_Плата*0.3,IIf(Sum(Робота.Тривалість)>50 And
Sum(Робота.Тривалість)<60,Працівник.Заробітна_Плата*0.2,Пра
цівник.Заробітна_Плата*0.1)) AS Підсумок
FROM Працівник INNER JOIN Робота ON
Працівник.[ID_Працівника] = Робота.[ID_Працівника]
GROUP BY Працівник.ПІБ_Працівника,
Працівник.Заробітна_Плата
HAVING (((Sum(Робота.Тривалість))>30))
ORDER BY
IIf(Sum(Робота.Тривалість)>=60,Працівник.Заробітна_Плата*0.3,
IIf(Sum(Робота.Тривалість)>30 And
Sum(Робота.Тривалість)<60,Працівник.Заробітна_Плата*0.2,Пра
цівник.Заробітна_Плата*0.1)) DESC;

```

2.25 Запропонувати варіант реалізації запиту «Статистична інформація про дохід працівників»

Запит «Статистична інформація про дохід працівників» виконує маніпуляції з даними таблиць «Працівник» та «Робота». Даний запит підраховує мінімальну, максимальну та середні вартості виконаних робіт. Також за допомогою мови SQL було проведено обрахунки для отримання кількості виконаних робіт і всього робочих днів. У кінцевому полі проводяться обрахунки обсягу середньої заробітної плати працівника за день. За допомогою функції UCase всю інформацію виводимо великими літерами.

```

SELECT UCase(Працівник.ПІБ_Працівника) AS Працівник,
Min(Робота.Вартість_Роботи) AS [Мінімальна вартість],
Max(Робота.Вартість_Роботи) AS [Максимальна вартість],
Avg(Робота.Вартість_Роботи) AS [Середня вартість],
Count(Робота.Тривалість) AS [Виконано одиниць роботи],
Sum(Робота.Тривалість) AS [Всього робочих днів],
Sum(Робота.Вартість_Роботи)/Sum(Робота.Тривалість) AS
[Середнє за день]
FROM Працівник INNER JOIN Робота ON Працівник.ID_Працівника
= Робота.ID_Працівника
GROUP BY Працівник.ПІБ_Працівника;

```

2.26 Запропонувати варіант реалізації запиту «Ціна портфеля криптовалюти»

Припустимо, що клієнт має певну кількість різної криптовалюти. Потрібно розрахувати загальну вартість портфеля криптовалюти враховуючи, що кожна криптовалюта має свій курс.

```
SELECT Криптогаманець.Логін, Sum([Кількість_монет]*[Курс]) AS  
[Ціна портфеля]  
FROM Криптовлота INNER JOIN Криптогаманець ON  
Криптовлота.Назва_валюти = Криптогаманець.Назва_валюти  
GROUP BY Криптогаманець.Логін  
ORDER BY Sum([Кількість_монет]*[Курс]) DESC;
```

2.27 Запропонувати варіант реалізації запиту «Прибуток гаманця в гривні»

Запит «Прибуток гаманця в гривні» показує, який прибуток отримав користувач за рахунок зміни курсу валют за останню добу. В запиті використовуються функції: CDbI для перетворення в числовий тип даних з плаваючою комою, Round для заокруглення отриманого числа.

```
SELECT Криптогаманець.Логін, Криптогаманець.Адреса_гаманця,  
Криптовлота.Зміна_курсу,  
Round(([Криптовлота].[Курс]*[Криптогаманець].[Кількість_монет]  
]*CDbI([Криптовлота].[Зміна_курсу])*27.35),2) AS  
Прибуток_гаманця  
FROM Криптовлота INNER JOIN Криптогаманець ON  
Криптовлота.Назва_валюти = Криптогаманець.Назва_валюти  
ORDER BY  
Round(([Криптовлота].[Курс]*[Криптогаманець].[Кількість_монет]  
]*CDbI([Криптовлота].[Зміна_курсу])*27.35),2);
```

2.28 З таблиці Покупки вивести всі записи, де Дата_придбання є травнем будь-якого року

```
SELECT *  
FROM Покупки  
WHERE MONTH(Дата_придбання) = 5;
```

2.29 З таблиці Покупки вивести всі записи, де Дата_придбання є 10-им числом будь-якого місяця і року

```
SELECT *  
FROM Покупки  
WHERE DAY(Дата_придбання) = 10;
```

2.30 З таблиці Покупки вивести всі записи, де Дата_придбання є 15-е серпня будь-якого року

```
SELECT *
FROM Покупки
WHERE MONTH(Дата_придбання) = 8 AND
      DAY(Дата_придбання) = 15;
```

2.31 З таблиці Покупки вивести всі записи, де Дата_придбання була понеділком

Для реалізації цього запиту потрібно застосувати функцію DAYOFWEEK, яка повертає ціле число від 1 до 7, причому відлік ведеться, починаючи з неділі. Тому у випадку, коли дата була понеділком, ця функція повертає значення 2.

```
SELECT *
FROM Покупки
WHERE WEEKDAY (Дата_придбання) = 2;
```

Аргументом цієї функції може бути і конкретно вказана дата. Так, для дати 21 травня 2021 року функція поверне значення 6, тобто, це була п'ятниця.

```
SELECT WEEKDAY ('2021-05-21');
```

2.32 Вибрати з таблиці Співробітники поля ПІБ_співробітника та Дата_народження, але записати в окремі поля день, місяць та рік народження співробітників

```
SELECT ПІБ_співробітника, DAY(Дата_народження) AS day,
      MONTH(Дата_народження) AS month,
      YEAR(Дата_народження) AS year
FROM Співробітники;
```

2.33 Запропонувати варіант реалізації запиту «Отримати покупки та їх вартість в поточному місяці від заданого фермера»

Запит «Отримати покупки та їх вартість в поточному місці від заданого фермера» виконує операцію вибору даних з таблиць «Покупки» та «Фермер». Такий запит відбирає дані за заданими параметрами, а саме, якщо покупки становили більше 2000 грн, придбання товару відбулось в поточному місяці, для введення імені фермера створено поле з ввідним параметром.

```
SELECT Фермер.ID_Фермера, Покупки.Назва_Товара,
```

```

        Покупки.Кількість, Покупки.Дата_Придбання,
        Покупки.Вартість
FROM Фермер INNER JOIN Покупки ON Фермер.[ID_Фермера] =
        Покупки.[ID_Фермера]
WHERE (((Фермер.ID_Фермера) Like "*" & [Введіть ПІБ] & "*")
AND ((Покупки.Вартість)>2000)
AND ((Year([Дата_Придбання])=Year(Now()))
AND ((Month([Дата_Придбання])=Month(Now())));

```

2.34 Запропонувати варіант запиту «Витрати по магазинах за останній місяць»

Запит «Витрати по магазинах за останній місяць» аналізує таблицю «Покупки», з неї виконується вибірка усіх різних магазинів, у яких відбувалося придбання товару. Підраховуються загальні витрати в кожному магазині та виконується відбір покупок тільки за останні тридцять днів.

```

SELECT DISTINCT (Покупки.Продавець) AS Продавець,
        Sum(Покупки.Вартість) AS [Вартість усіх куплених
товарів]
FROM Покупки
WHERE (((Покупки.Дата_Придбання) Between Date() And
        DateAdd("m",-1,Date()))
GROUP BY Покупки.Продавець
ORDER BY Sum(Покупки.Вартість);

```

Щоб додати дні до дати, можна використовувати день року («у»), день («d») або будні дні («w»), а операції з роком здійснюються через параметр «уууу».

3 ТЕСТОВІ ЗАВДАННЯ 1

1. Що таке база даних?

- 1) будь-який текстовий файл,
- 2) організована структура для зберігання інформації,
- 3) будь-яка інформація, що подана у табличній формі,
- 4) будь-яка електронна таблиця.

2. Яка з перерахованих властивостей є властивістю реляційної бази даних?

- 1) декілька вузлів рівня, пов'язані з вузлом одного рівня,
- 2) порядок слідування рядків в таблиці довільний,
- 3) кожен стовпець має унікальне ім'я,
- 4) для кожної таблиці можна визначити первинний ключ.

3. Що таке SQL?

- 1) мова розмітки бази даних,
- 2) структурована мова запитів,
- 3) мова програмування низького рівня,
- 4) мова програмування високого рівня.

4. Яка база даних будується на основі таблиць і тільки таблиць?

- 1) мережна,
- 2) ієрархічна,
- 3) реляційна,
- 4) ніяка.

5. Який з нижчеперерахованих елементів не є об'єктом MS Access?

- 1) таблиця,
- 2) книга,
- 3) запит,
- 4) звіт.

6. В якій моделі баз даних існують горизонтальні та вертикальні зв'язки між елементами?

- 1) мережній,
- 2) ієрархічній,
- 3) реляційній,
- 4) об'єктно-орієнтованій.

7. Який з нижчеперерахованих запитів не можна побудувати?

- 1) простий,
- 2) перехресний,
- 3) на створення таблиці,
- 4) паралельний.

8. Що таке поле?

- 1) стовпець в таблиці,
- 2) вікно конструктора,
- 3) текст будь-якого розміру,
- 4) рядок в таблиці.

9. Що таке запит?

- 1) вікно конструктора,
- 2) зв'язана таблиця,
- 3) головна таблиця,
- 4) засіб відбору даних.

10. В чому полягає функція ключового поля?

- 1) однозначно визначити таблицю,
- 2) однозначно визначити запис,
- 3) визначати заголовок стовпця таблиці,
- 4) вводить обмеження для перевірки правильності введення даних.

11. З чого складається макрос?

- 1) з набору тегів,
- 2) з сукупності операторів Visual Basic,
- 3) з набору гіперпосилань,
- 4) з набору мікрокоманд.

12. Якого розділу не існує в конструкторі форм?

- 1) заголовка,
- 2) верхнього колонтитула,
- 3) області даних,
- 4) загального.

13. Таблиці, запити, звіти, форми – це:

- 1) єдиний файл БД,
- 2) окремі файли, вміщені в папку,
- 3) щось інше,
- 4) папки.

14. Для створення нової таблиці потрібно:

- 1) активізувати команди **Файл, Создать**,
- 2) відкрити вкладку **Таблицы**, активізувати кнопку **Создать**,
- 3) після завантаження **MS Access** активізувати перемикач,
- 4) зберегти створений файл.

15. В режимі конструктора таблиці можна виконувати дії:

- 1) додати нове поле,

- 2) додати нове значення поля,
- 3) встановити зв'язок між таблицями,
- 4) вносити дані в таблицю.

16. Змінити формат числового поля таблиці можна:

- 1) в режимі таблиці,
- 2) в конструкторі таблиць,
- 3) і в режимі таблиці, і в режимі конструктора таблиць,
- 4) під час внесення даних в таблицю.

17. Для встановлення необхідної кількості десяткових знаків числового поля потрібно:

- 1) змінити значення параметра **Число десятичних знаків**,
- 2) змінити значення параметрів **Размер поля**, **Формат поля**, **Число десятичних знаків**,
- 3) ввести необхідну кількість знаків при створенні значення,
- 4) вибрати тип поля.

18. Маску виведення для поля типу Дата/Время можна встановити за допомогою параметра:

- 1) розмір поля,
- 2) формат поля,
- 3) маска введення,
- 4) фіксованого числового формату.

19. В текстовому полі можна зберігати:

- 1) тільки літери та символи,
- 2) літери, цифри,
- 3) спеціальні символи, пропуски,
- 4) символічні та числові дані.

20. Ім'я поля таблиці має довжину:

- 1) до 64 символів,
- 2) до 8 символів,
- 3) іншу кількість символів,
- 4) 256 символів.

21. Над полем типу Дата/Время можна виконувати:

- 1) тільки перегляд значень,
- 2) додавання,
- 3) віднімання,
- 4) порівняння.

22. Ключове поле таблиці може мати:

- 1) повторювані символи,

- 2) десяткові значення чисел,
- 3) значення типу **Дата/Время**,
- 4) назву поля.

23. Майстер підстановок використовується:

- 1) для створення полів зі значеннями,
- 2) для створення полів зі значеннями, що повторюються,
- 3) для додавання значень полів з інших таблиць,
- 4) при розрахунку функцій.

24. Для вилучення поля з таблиці потрібно:

- 1) виділити поле в режимі таблиці, натиснути клавішу **Del**,
- 2) виділити поле в режимі таблиці,
- 3) виділити поле в конструкторі таблиць, натиснути клавішу **Del**,
- 4) виконати інші дії.

25. Запити використовуються для:

- 1) виведення значень таблиці у вигляді, зручному для користувача,
- 2) відбору значень відповідно до поставленої умови,
- 3) створення набору вкладок,
- 4) введення значень таблиці у вигляді, зручному для користувача.

26. Кнопки для пошуку потрібних значень у таблицях можна використати для:

- 1) пошуку значень тільки одного поля,
- 2) пошуку значень в іншій БД,
- 3) пошуку значень будь-якого поля, в якому встановлено курсор,.
- 4) пошуку значень декількох полів

27. Для розрахунку проміжних підсумків у звітах необхідно:

- 1) мати поля зі значеннями, що повторюються,
- 2) мати числові поля у звіті,
- 3) у конструкторі в області даних створити відповідне розрахункове поле,
- 4) мати відповідне поле.

28. Підпорядкована форма створюється для:

- 1) надання користувачеві додаткової інформації,
- 2) проведення розрахунків,
- 3) вміщення командних кнопок,
- 4) внесення даних.

29. Де виконується розрахунок загальних функцій у звітах:

- 1) в області **Примечание отче та**,
- 2) в області **Примечание группы**,

- 3) області даних,
- 4) області таблиць.

30. Що таке об'єкт БД:

- 1) інформація,
- 2) елемент інформаційної системи,
- 3) суть,
- 4) атрибут.

31. Які типи зв'язку не існують в БД:

- 1) 1: 1,
- 2) 1: N,
- 3) M : N,
- 4) 1 : 0.

32. Частина реальної системи – це:

- 1) атрибут,
- 2) суть,
- 3) предметна область,
- 4) область значень.

33. Ключовим елементом даних є:

- 1) елемент, що визначає первинний ключ,
- 2) елемент, що визначає значення інших елементів,
- 3) елемент, що визначає альтернативний ключ,
- 4) елемент, що визначає кортеж.

34. Клас належності буває:

- 1) обов'язковим,
- 2) проміжним,
- 3) необов'язковим,
- 4) середнім.

35. Схема даних пов'язує:

- 1) форми,
- 2) звіти,
- 3) таблиці,
- 4) запити.

36. Які з перерахованих нормальних форм вам відомі:

- 1) 1НФ,
- 2) 2НФ,
- 3) 3НФ,
- 4) 4НФ.

37. Які операції над множинами вам відомі:

- 1) об'єднання,
- 2) переріз,
- 3) доповнення,
- 4) різниця.

38. Що таке СУБД:

- 1) схема управління базами даних,
- 2) система управління базами даних,
- 3) схема управління банками даних,
- 4) система управління банками дани.

39. Кожна таблиця для введення інформації має:

- 1) запит,
- 2) звіт,
- 3) додаткову таблицю,
- 4) форму.

40. Впорядкувати дані в таблиці можна:

- 1) за зростанням значень,
- 2) за спаданням значень,
- 3) за алфавітом,
- 4) як заманеться.

ВІДПОВІДІ

№ тест, завдання	Варіанти відповіді				№ тест, завдання	Варіанти відповіді			
	1	2	3	4		1	2	3	4
1	+				21	+			
2	+				22				+
3	+				23			+	
4	+				24	+			
5	+				25		+		
6	+				26			+	
7	+				27			+	
8	+				28	+			
9	+				29	+			
10	+				30		+		
11	+				31				+
12	+				32			+	
13	+				33		+		
14	+				34	+		+	
15	+				35			+	
16	+				36	+	+	+	
17	+				37	+	+	+	+
18	+				38		+		
19	+				39				+
20	+				40	+	+		

ТЕСТИ З РОЗДІЛУ

Реляційні бази даних (БД). СУБД. Основи клієнт-серверної технології
Мова SQL.

Розробник Романюк О. Н.

4. ТЕСТОВІ ЗАВДАННЯ 2

Ч.ч.	Запитання	Відповіді
1	Структури даних – це ...	<p>a) структурно організовані сукупності даних;</p> <p>b) сукупності даних;</p> <p>c) послідовно організовані сукупності даних;</p> <p>d) паралельно організовані сукупності даних.</p>
2	Предметна (проблемна) область – це ...	<p>a) сукупність об'єктів управління (або їх частин), інформація про які моделюється за допомогою баз даних і використовується для розв'язування відповідних функціональних задач;</p> <p>b) сукупність об'єктів управління (або їх частин), інформація про які перетворюється за допомогою баз даних і використовується для розв'язування відповідних функціональних задач;</p> <p>c) управління (або їх частин), інформація про яке моделюється за допомогою баз даних і використовується для розв'язування відповідних функціональних задач;</p> <p>d) інформація яка моделюється за допомогою баз даних і використовується для розв'язування відповідних функціональних задач.</p>
3	Базою даних (БД) називається ...	<p>a) сукупність взаємопов'язаних структурованих даних, що належать до певної проблемної області, які зберігаються в пам'яті комп'ютера так, щоб забезпечити подальше розширення (нарощування), поновлення (актуалізацію) даних відповідно до вибраної або існуючої структури відношень (взаємозв'язків між даними);</p> <p>b) сукупність даних, що належать до певної проблемної області, які зберігаються в пам'яті комп'ютера так, щоб забезпечити подальше розширення (нарощування), поновлення (актуалізацію) даних відповідно до вибраної або існуючої структури відношень (взаємозв'язків між даними);</p> <p>c) сукупність систем, що належать до певної проблемної області, які зберігаються в пам'яті комп'ютера так, щоб забезпечити подальше розширення (нарощування), поновлення (актуалізацію) даних відповідно до вибраної або існуючої структури відношень (взаємозв'язків між даними).</p> <p>d) сукупність взаємопов'язаних структурованих даних, які зберігаються в пам'яті комп'ютера так, щоб забезпечити подальше розширення (нарощування), поновлення (актуалізацію) даних.</p>

4	Системою управління базою даних (СУБД) називається ...	<p>а) спеціальна мова опису і управління базами даних (БД), яка забезпечує уніфікованість засобів доступу до накопиченої в БД інформації та незалежність даних від програм, які ці дані використовують;</p> <p>б) спеціальний засіб управління базами даних (БД), який забезпечує уніфікованість доступу до накопиченої в БД інформації та незалежність даних від програм, які ці дані використовують;</p> <p>в) спеціальна мова опису і управління базами даних (БД), яка забезпечує уніфікованість доступу до накопиченої в БД інформації;</p> <p>г) структура даних, які забезпечує уніфікованість доступу до накопиченої інформації та незалежність даних від програм, які ці дані використовують.</p>
5	Банк даних (система баз даних) – це ...	<p>а) сукупність незалежних і не пов'язаних між собою баз даних, що керуються засобами однієї СУБД;</p> <p>б) сукупність залежних і пов'язаних між собою баз даних, що керуються засобами однієї СУБД;</p> <p>в) сукупність незалежних і не пов'язаних між собою баз даних;</p> <p>г) сукупність баз даних, що керуються засобами однієї СУБД.</p>
6	Властивості баз даних.	<p>а) 1) ненадлишковість або мінімально необхідна надлишковість; 2) незалежність від програм користувача, які ці дані використовують; 3) захищеність; 4) надійність;</p> <p>б) 1) надлишковість; 2) залежність від програм користувача, які ці дані використовують; 3) захищеність; 4) надійність;</p> <p>в) 1) ненадлишковість або мінімально необхідна надлишковість; 2) незалежність від програм користувача, які ці дані використовують; 3) захищеність; 4) ненадійність.</p> <p>г) 1) надлишковість, 2) незалежність від програм користувача, які ці дані використовують, 3) незахищеність, 4) надійність.</p>
7	Запис – це ...	<p>а) елемент збереження інформації у базі даних, що містить сукупність відомостей (характеристик, атрибутів) про окремий об'єкт предметної області бази даних;</p> <p>б) елемент збереження інформації у базі даних;</p> <p>в) елемент збереження інформації у файлі, що містить сукупність відомостей (характеристик, атрибутів) про окремий об'єкт предметної області бази даних.</p> <p>г) елемент збереження інформації у банку даних, що містить сукупність відомостей (характеристик, атрибутів).</p>

8	Набір даних – це ...	<p>a) екземпляр іменованої сукупності записів, що має просту структуру дворівневого дерева;</p> <p>b) послідовність записів, що має просту структуру дворівневого дерева;</p> <p>c) запис, що має просту структуру дворівневого дерева;</p> <p>d) екземпляр іменованої сукупності записів.</p>
9	Тип набору даних ...	<p>a) визначає відношення між типами записів, один з яких виділяється як запис-власник (вузол верхнього рівня), а решта записів - як елементи цього набору, які являють собою вузли нижнього рівня;</p> <p>b) визначає відношення між типами записів, один з яких виділяється як запис-власник (вузол верхнього рівня);</p> <p>c) визначає відношення між типами файлів, один з яких виділяється як файл-власник (вузол верхнього рівня), а решта файлів – як елементи цього набору, які являють собою вузли нижнього рівня;</p> <p>d) визначає розмір записів, один з яких виділяється як запис-власник (вузол верхнього рівня), а решта записів – як елементи цього набору, які являють собою вузли нижнього рівня.</p>
10	Таблиця – це ...	<p>a) це іменована впорядкована структура, що складається з кінцевого набору однотипних записів;</p> <p>b) іменована сукупність різнотипних записів;</p> <p>c) іменована сукупність різноманітних записів;</p> <p>d) неіменована сукупність різноманітних записів.</p>
11	Кортеж – це ...	<p>a) рядок таблиці-відношення;</p> <p>b) колонка таблиці-відношення;</p> <p>c) таблиця-відношення;</p> <p>d) колонка таблиці-відношення.</p>
12	Відношення – це ...	<p>a) екземпляр набору даних, поданий у вигляді таблиці або відповідною матрицею;</p> <p>b) набір даних, поданий у вигляді таблиці або відповідною матрицею;</p> <p>c) екземпляр набору даних, поданий у вигляді списку;</p> <p>d) екземпляр набору даних, поданий у вигляді графіка.</p>
13	Степінь відношення – це ...	<p>a) кількість полів (колонок) у таблиці-відношенні;</p> <p>b) кількість рядків у таблиці-відношенні;</p> <p>c) кількість даних у таблиці-відношенні;</p> <p>d) кількість полів (колонок) у списку.</p>
14	Бінарне відношення (реляція) ...	<p>a) відношення степеня 2;</p> <p>b) сума відношень степеня 2;</p> <p>c) різниця відношень степеня 2;</p> <p>d) відношення степеня 4.</p>

15	Тернарне відношення (реляція) ...	<p>a) відношення степеня 3;</p> <p>b) добуток відношень степеня 3;</p> <p>c) відношення множників степеня 3;</p> <p>d) відношення записів степеня 3.</p>
16	n-арне відношення (реляція) ...	<p>a) відношення степеня n;</p> <p>b) кут відношення степеня n;</p> <p>c) степінь n;</p> <p>d) попереднє відношення степеня n.</p>
17	Процес нормалізації – це ...	<p>a) процес переходу від деякої складної структури даних до дворівневих структур за умови забезпечення однозначності пошуку всіх елементів даних;</p> <p>b) процес переходу від деякої складної форми даних до дворівневих форм за умови забезпечення однозначності пошуку всіх елементів даних;</p> <p>c) процес переходу від деякої простої структури даних до багаторівневих структур за умови забезпечення однозначності пошуку всіх елементів даних;</p> <p>d) процес пошуку від деякої складної структури даних до дворівневих структур за умови забезпечення однозначності подання всіх елементів даних.</p>
18	Перша нормальна форма (1НФ)	<p>a) всі дані подаються двоимірними таблицями-відношеннями з колонками найпростішої структури;</p> <p>b) окремі дані подаються двовимірними схемами-відношеннями з колонками найпростішої структури;</p> <p>c) всі дані подаються багатовимірними таблицями-відношеннями з рядками найпростішої структури;</p> <p>d) всі дані подаються колонками найпростішої структури.</p>
19	Друга нормальна форма (2НФ)	<p>a) подання даних відповідає 1НФ, але кожний атрибут, що є первинним або елементарним, тобто, може бути первинним ключем (індексом), повністю залежить від всього ключа відношення, в загальному випадку складеного з усіх можливих ключів (індексів);</p> <p>b) перехід даних відповідає 1 С, але кожний атрибут, що є первинним або елементарним, тобто, може бути первинним ключем (індексом), повністю залежить від всього ключа відношення, в загальному випадку складеного з усіх можливих ключів (індексів);</p> <p>c) подання даних не відповідає 1НФ, але кожний атрибут, що є вторинним або елементарним, тобто, може бути первинним ключем (індексом), повністю залежить від всього ключа відношення, в загальному випадку складеного з усіх можливих ключів (індексів);</p> <p>d) подання даних відповідає 1НФ, але кожний атрибут, що є первинним або складним, тобто, може бути первинним ключем (показником), повністю залежить від всього ключа відношення, в окремому випадку складеного з усіх можливих ключів (показників).</p>

20	Третя нормальна форма (3НФ)	<p>a) подання даних відповідає всім вимогам 1НФ і 2НФ, але кожен атрибут, що не є первинним, безпосередньо (нетранзитивно) і однозначно залежить від всього ключа відношення. В 3НФ елементи даних в логічному записі залежать лише від ключа цього запису.</p> <p>b) подання даних відповідає всім вимогам 1НФ і 2НФ, але кожен атрибут, що не є вторинним, безпосередньо (нетранзитивно) і однозначно залежить від всього ключа відношення. В 3НФ елементи даних в нелогічному записі залежать лише від ключа цього запису;</p> <p>c) подання даних не відповідає всім вимогам 1НФ і 2НФ, але кожен атрибут, що не є первинним, опосередковано (нетранзитивно) і однозначно залежить від всього ключа відношення. В 3НФ елементи даних в логічному записі залежать не лише від ключа цього запису;</p> <p>d) перехід даних відповідає всім вимогам 1НФ і 2НФ, але кожен компонент, що не є первинним, безпосередньо (транзитивно) і однозначно залежить від всього ключа відношення. В 3НФ елементи даних в логічному записі залежать лише від ключа цього запису.</p>
21	Домен – це ...	<p>a) набір значень одного стовпця (колонки) таблиці-відношення одного типу;</p> <p>b) склад значень одного стовпця (колонки) схеми-відношення одного типу;</p> <p>c) набір значень одного стовпця (колонки) графіка одного типу;</p> <p>d) набір понять одного рядка таблиці-відношення одного виду.</p>
22	Схема даних (концепту-альна модель бази даних) – це ...	<p>a) загальна схема, що описує логічну структуру бази даних (БД) і складається експертами в даній галузі знань. Вона являє собою таблицю типів даних, що використовуються і містить імена об'єктів і їх атрибути. Схема даних об'єктивно відображає відношення між об'єктами, що взаємодіють і не змінюється з часом, а лише уточнюється;</p> <p>b) загальна схема, що описує логічну форму бази даних (БД) і складається експертами в даній галузі знань. Вона являє собою схему типів даних, що використовуються і містить імена об'єктів і їх атрибути. Схема даних об'єктивно відображає різницю між об'єктами, що взаємодіють і не змінюється з часом, а лише уточнюється;</p> <p>c) загальна схема, що описує логічну структуру бази даних (БД) і складається експертами в даній галузі знань. Вона являє собою таблицю типів даних, що використовуються, і містить опис об'єктів і їх компоненти. Схема даних об'єктивно відображає відношення між об'єктами, що взаємодіють, і не змінюється з часом, а лише примножується;</p> <p>d) таблиця, що описує склад бази даних (БД) і складається експертами в даній галузі знань. Вона являє собою таблицю типів даних, що використовуються, і містить імена суб'єктів і їх атрибути. Схема даних об'єктивно відображає відношення між суб'єктами, що взаємодіють, і змінюється з часом, уточнюється.</p>

23	Підсхема даних – це	<p>a) часткова схема даних, яку складає користувач під час роботи з базою даних, і яка являє собою лише деякий екземпляр (зразок, варіант) схеми даних із певними значеннями елементів даних;</p> <p>b) часткова таблиця даних, яку складає користувач під час роботи з базою даних, і яка являє собою лише деякий екземпляр (зразок, варіант) таблиці даних із певними значеннями часток даних;</p> <p>c) часткова схема даних, яку складає користувач під час роботи з базою даних, і яка являє собою лише конкретний екземпляр (зразок, варіант) схеми даних зі зразками елементів таблиці;</p> <p>d) загальна схема даних, яку складає користувач під час роботи з базою даних, і яка являє собою лише набір даних із певними значеннями елементів даних.</p>
24	Компонент представлення (presentation) ...	<p>a) виконує функції введення та відображення даних;</p> <p>b) виконує функції обробки даних;</p> <p>c) виконує функції архівування даних;</p> <p>d) виконує функції оптимізації даних.</p>
25	Прикладний компонент (business application) ...	<p>a) реалізує прикладні функції, характерні для даної предметної області;</p> <p>b) реалізує прикладні функції, не характерні для даної предметної області;</p> <p>c) перехоплює прикладні функції, характерні для даної предметної області;</p> <p>d) аналізує прикладні функції, характерні для даної предметної області.</p>
26	Ієрархічна модель даних ...	<p>a) будується за принципом ієрархії, тобто, один об'єкт є головним, а інші – підлеглими. В ієрархічній БД встановлюється відношення підпорядкування типу «1: ∞» між головним і підлеглими об'єктами;</p> <p>b) ґрунтується за принципом ієрархії, тобто, один об'єкт є першим, а інші – вторинними. В ієрархічній БД встановлюється відношення підпорядкування типу «1: ∞» між першим і вторинними об'єктами;</p> <p>c) будується за схемою ієрархії, тобто, один об'єкт є головним, а інші – підлеглими. В ієрархічній БД відмінюються відношення підпорядкування типу «1: ∞» між головним і підлеглими об'єктами;</p> <p>d) будується за принципом структуризації, тобто, один об'єкт є структурною одиницею. В структурній БД встановлюється відношення підпорядкування типу «∞:1» між головним і підлеглими об'єктами.</p>

27	Реляційна модель даних ...	<p>a) подає об'єкти і відношення між ними у вигляді таблиць, а всі операції над об'єктами виконуються як операції над таблицями. При цьому відношення (зв'язки) також розглядаються як об'єкти;</p> <p>b) надає об'єкти і різницю між ними у вигляді таблиць, а всі операції над об'єктами виконуються як операції над таблицями. При цьому відношення (зв'язки) також розглядаються як об'єкти;</p> <p>c) подає об'єкти і відношення між ними у вигляді графіків, а всі операції над об'єктами проходять як операції над схемами. При цьому відношення (зв'язки) також розглядаються як об'єкти;</p> <p>d) подає об'єкти і відношення між ними у вигляді таблиць, а всі переходи виконуються як операції над таблицями. При цьому відношення (зв'язки) також розподіляються як об'єкти.</p>
28	Сіткова модель даних ...	<p>a) будується за принципом «головний і підлеглий тип одночасно», тобто, будь-який тип даних одночасно може породжувати декілька підлеглих типів (бути власником набору даних) і в той же самий час бути підпорядкованим для декількох головних (бути елементом інших) елементів даних;</p> <p>b) ґрунтується за принципом «головний і підлеглий тип одночасно», тобто, будь-який тип даних одночасно може попереджати декілька підлеглих типів (бути власником набору даних) і в той же самий час не бути підпорядкованим для декількох головних (бути елементом інших) елементів даних;</p> <p>c) будується за принципом «основний і підпорядкований тип одночасно», тобто, будь-який тип даних одночасно може породжувати п'ять підлеглих типів (бути власником набору даних) і в той же самий час бути підпорядкованим для декількох основних (бути елементом інших) елементів даних;</p> <p>d) спостерігається за принципом «головний і підлеглий тип в часі», тобто, будь-який тип даних може погоджувати декілька підлеглих типів (бути власником набору даних).</p>
29	Фактографічні бази даних ...	<p>a) використовують строго форматovanі записи однакової довжини, які містять в собі певну кількість полів даних. Пошук здійснюється за ключами при однозначній відповідності ключів і даних;</p> <p>b) використовують строго форматovanі схеми однакової довжини, які містять в собі певну кількість рядків даних. Пошук здійснюється за ключами при однозначній відповідності ключів і даних;</p> <p>c) використовують строго форматovanі типи однакової висоти, які містять в собі певну кількість полів даних. Перехід здійснюється за ключами при двозначній відповідності ключів і даних;</p> <p>d) забезпечують чітко форматovanі записи однакової ширини, які містять в собі певну кількість полів даних. Пошук здійснюється за ключами при однозначній невідповідності ключів і даних.</p>

30	Інформаційно-пошукові бази даних ...	<p>а) складаються з записів змінної довжини, а пошук документів здійснюється за їх змістом. Для цього використовуються визначальні (ключові) слова – дескриптори;</p> <p>б) виконують записи змінної довжини, а пошук документів здійснюється за їх розміром. Для цього використовуються визначальні (ключові) слова – корелятори;</p> <p>с) складаються з записів змінної ширини, а витяг з документів здійснюється за їх змістом. Для цього вико3 документів здійснюється за їх змістом. Для цього використовуються визначальні схеми.</p>
31	Семантичний підхід до моделювання даних предметної області полягає в ...	<p>а) інтелектуалізації банків даних і організації взаємодії користувача з базою даних на рівні уявлень про предметну область, а не на рівні структур даних;</p> <p>б) інтелектуалізації банків даних і організації взаємодії користувача з базою даних на рівні уявлень про предметну область, а не на рівні таблиць даних;</p> <p>с) інтелектуалізації банків даних і організації взаємодії користувача з базою даних на рівні моделей предметної області, а не на рівні структур даних;</p> <p>д) ноотропності бази даних і організації взаємодії користувача з банками даних на рівні уявлень про предметну область, а не на рівні її ділянок.</p>
32	Інфологічне моделювання полягає ...	<p>а) у виокремленні інформаційних об'єктів предметної області, а також у визначенні характеристик об'єктів і зв'язків між ними;</p> <p>б) в узагальненні інформаційних об'єктів предметної області, а також у визначенні характеристик об'єктів і зв'язків між ними;</p> <p>с) у виокремленні інформаційних схем предметної області, а також у визначенні характеристик бази даних;</p> <p>д) у визначенні конформаційних об'єктів предметної області, а також у визначенні характеристик об'єктів і зв'язків між ними.</p>
33	Даталогічне моделювання полягає у ...	<p>а) безпосередньому моделюванні даних з урахуванням специфіки конкретної СУБД;</p> <p>б) логічному моделюванні даних з урахуванням специфіки конкретної СУБД;</p> <p>с) структурному моделюванні даних з урахуванням специфіки конкретної СУБД;</p> <p>д) віртуальному моделюванні даних з урахуванням специфіки конкретної СУБД.</p>

34	Об'єкт предметної області являє собою елемент певної ...	<p>a) системи або предметної області, інформація про який зберігається у банку даних цієї предметної області. Об'єкт може бути реальним або абстрактним. Характеристиками об'єктів є їх атрибути;</p> <p>b) структури, інформація про яку зберігається у банку даних цієї структури. Об'єкт може бути реальним або абстрактним. Характеристиками об'єктів є їх атрибути;</p> <p>c) системи або предметної області. Об'єкт може бути реальним або абстрактним. Характеристиками об'єктів є їх розміри;</p> <p>d) моделі, інформація про яку зберігається у банку даних цієї моделі. Об'єкт може бути реальним або абстрактним. Характеристиками об'єктів є їх атрибути.</p>
35	Ключ – це ...	<p>a) поле або набір полів запису, значення яких однозначно визначають запис у базі даних»</p> <p>b) кортеж запису, значення якого однозначно визначають запис у базі даних;</p> <p>c) рядок запису, значення якого однозначно визначають запис у базі даних;</p> <p>d) рядок або набір рядків запису, значення яких однозначно визначають запис у базі даних.</p>
36	Пряма ідентифікація об'єктів полягає у безпосередньому ...	<p>a) вказуванні імен об'єктів;</p> <p>b) копіюванні імен об'єктів;</p> <p>c) дублюванні імен об'єктів;</p> <p>d) вказуванні атрибутів об'єктів.</p>
37	Непряма ідентифікація об'єктів наголошує на вказуванні ...	<p>a) властивостей (атрибутів) або ключів об'єкта;</p> <p>b) властивостей (атрибутів) об'єкта;</p> <p>c) ключів об'єкта;</p> <p>d) імені об'єкта.</p>
38	Концептуальна схема предметної області це ...	<p>a) множина всіх типів, яка має певну структуру. Структура на множині типів задається множиною відношень і операцій, визначеними на множині типів.</p> <p>b) множина деяких типів, яка має певну структуру. Структура на множині типів задається множиною відношень і операцій, визначеними на множині типів.</p> <p>c) множина простих типів, яка має певну структуру. Структура на множині типів задається множиною відношень і операцій, визначеними на множині типів.</p> <p>d) множина всіх типів, яка не має певної структури.</p>

39	Реляційна модель даних визначається правилами ...	<p>a) 1) інформації; 2) гарантованого доступу; 3) підтримки недійсних значень; 4) динамічного каталогу; 5) вичерпної підмови даних; 6) оновлення подання; 7) додавання даних; 8) незалежності фізичних даних; 9) незалежності логічних даних; 10) незалежності умов цілісності; 11) незалежності поширення (розповсюдження); 12) унікальності мови даних;</p> <p>b) 1) інформації; 2) гарантованого доступу; 3) підтримки недійсних значень; 4) статичного каталогу; 5) вичерпної підмови даних; 6) оновлення подання; 7) додавання даних; 8) незалежності фізичних даних; 9) незалежності логічних даних; 10) незалежності умов цілісності; 11) незалежності поширення (розповсюдження); 12) унікальності мови даних;</p> <p>c) 1) інформації; 2) гарантованого доступу; 3) підтримки недійсних значень; 4) динамічного каталогу; 5) вичерпної підмови даних; 6) оновлення подання; 7) додавання даних; 8) залежності фізичних даних; 9) залежності логічних даних; 10) залежності умов цілісності; 11) залежності поширення (розповсюдження); 12) унікальності мови даних;</p> <p>d) 1) інформації; 2) гарантованого доступу; 3) підтримки дійсних значень; 4) віртуального каталогу; 5) вичерпної підмови даних; 6) оновлення подання; 7) видалення даних; 8) незалежності фізичних даних; 9) незалежності логічних даних; 10) незалежності умов цілісності; 11) незалежності поширення (розповсюдження); 12) унікальності мови даних.</p>
40	Зовнішній ключ – це ...	<p>a) колонка однієї таблиці, значення в якій збігаються зі значенням колонки, що є первинним ключем в іншій таблиці;</p> <p>b) рядок однієї таблиці, значення в якому збігаються зі значенням колонки, що є первинним ключем в іншій таблиці;</p> <p>c) колонки всіх таблиць, значення в яких збігаються зі значенням колонки;</p> <p>d) таблиця, значення в якій збігаються зі значенням, що є первинним ключем в іншій таблиці.</p>
41	Складний ключ – це ...	<p>a) первинний ключ, що складається з декількох полів, тобто значень, що містяться в різних колонках таблиці;</p> <p>b) ключ, що складається з декількох полів, тобто значень, що містяться в різних колонках таблиці;</p> <p>c) первинний ключ, що складається з декількох записів, тобто значень, що містяться в різних рядках таблиці;</p> <p>d) зовнішній ключ, що складається з декількох полів, тобто значень, що містяться в різних колонках таблиці.</p>
42	SQL – це ...	<p>a) структурована мова запитів для обробки даних у реляційних базах даних;</p> <p>b) стандартна мова запитів для обробки даних у ієрархічних базах даних;</p> <p>c) структурована мова запитів для обробки даних у мережних базах даних;</p> <p>d) мова запитів для обробки даних у базах даних.</p>

43	Можливості SQL:	<p>a) 1) організація даних, 2) читання даних, 3) безпосередня обробка даних, 4) управління доступом, 5) спільне використання даних, 6) підтримання цілісності даних;</p> <p>b) 1) організація даних, 2) захист даних, 3) безпосередня архівація даних, 4) управління доступом, 5) спільне використання даних, 6) підтримання цілісності даних;</p> <p>c) 1) організація даних, 2) читання даних, 3) безпосереднє збереження даних, 4) управління доступом, 5) спільне використання даних, 6) підтримання цілісності даних;</p> <p>d) 1) організація даних, 2) читання даних, 3) безпосередня обробки даних, 4) управління доступом, 5) монопольне використання даних.</p>
44	Переваги SQL:	<p>a) 1) незалежність від СУБД, 2) переносність між різними обчислювальними системами, 3) стандартизація, 4) підтримання найбільшими виробниками програмного забезпечення, 5) реляційна мова, 6) високорівнева мова, аналогічна звичайній англійській мові, 7) можливість виконання спеціальних високорівневих інтерактивних запитів, 8) забезпечення програмного доступу до бази даних, 9) можливість різноманітного подання даних, 10) підтримка архітектури клієнт-сервер;</p> <p>b) 1) незалежність від СУБД, 2) переносність між різними обчислювальними системами, 3) стандартизація, 4) підтримання найбільшими виробниками програмного забезпечення, 5) реляційна мова, 6) високорівнева мова, аналогічна звичайній англійській мові;</p> <p>c) 1) залежність від СУБД, 2) переносність між різними обчислювальними системами, 3) стандартизація, 4) підтримання найбільшими виробниками програмного забезпечення. 5) реляційна мова, 6) високорівнева мова, аналогічна звичайній англійській мові, 7) можливість виконання спеціальних високорівневих інтерактивних запитів, 8) забезпечення програмного доступу до бази даних;</p> <p>d) 1) залежність від СУБД, 2) переносність між різними обчислювальними системами, 3) стандартизація, 4) підтримання найбільшими виробниками програмного забезпечення, 5) реляційна мова, 6) низькорівнева мова, аналогічна звичайній англійській мові, 7) можливість виконання спеціальних високорівневих інтерактивних запитів, 8) забезпечення програмного доступу до бази даних, 9) можливість різноманітного подання даних, 10) підтримка архітектури клієнт-сервер.</p>
45	FS – це ...	<p>a) модель файлового сервера;</p> <p>b) модель віддаленого доступу до даних;</p> <p>c) модель сервера прикладних задач;</p> <p>d) модель сервера баз даних.</p>
46	RDA – це ...	<p>a) модель віддаленого доступу до даних;</p> <p>b) модель файлового сервера;</p> <p>c) модель сервера прикладних задач;</p> <p>d) модель сервера баз даних.</p>

47	DBS – це ...	<ul style="list-style-type: none"> a) модель сервера баз даних; b) модель файлового сервера; c) модель віддаленого доступу до даних; d) модель сервера прикладних задач.
48	AS – це ...	<ul style="list-style-type: none"> a) модель прикладного сервера (сервера прикладних задач); b) модель файлового сервера; c) модель віддаленого доступу до даних; d) модель сервера баз даних.
49	Серед інформаційних задач з позиції правил організації вирішення задач у встановленому управлінському режимі розрізняють:	<ul style="list-style-type: none"> a) регламентні і запитно-довідкові; b) систематичні, епізодичні, випадкові; c) термінові і нетермінові; d) типові і нетипові.
50	Серед інформаційних задач з регулярності вирішення розрізняють:	<ul style="list-style-type: none"> a) систематичні, епізодичні, випадкові; b) термінові і нетермінові; c) типові і нетипові; d) одиничні, групові, масові.
51	Серед інформаційних задач за терміном вирішення розрізняють:	<ul style="list-style-type: none"> a) термінові і нетермінові; b) систематичні, епізодичні, випадкові; c) типові і нетипові; d) одиничні, групові, масові.
52	Серед інформаційних задач з позиції технологічної спільності правил вирішення задач розрізняють:	<ul style="list-style-type: none"> a) типові і нетипові; b) термінові і нетермінові; c) систематичні, епізодичні, випадкові; d) одиничні, групові, масові.
53	Серед інформаційних задач за повторюваністю розрізняють:	<ul style="list-style-type: none"> a) одиничні, групові, масові; b) типові і нетипові; c) термінові і нетермінові; d) систематичні, епізодичні, випадкові.
54	Серед інформаційних задач за математичною основою обчислень розрізняють:	<ul style="list-style-type: none"> a) задачі прямого розрахунку, оптимізаційні задачі, багатоваріантні задачі з пошуком найкращих рішень; b) типові і нетипові; c) термінові і нетермінові; d) систематичні, епізодичні, випадкові.
55	Серед інформаційних задач з позиції формалізації правил вирішення розрізняють:	<ul style="list-style-type: none"> a) формалізовані, неформалізовані; b) типові і нетипові; c) термінові і нетермінові; d) систематичні, епізодичні, випадкові.

56	Програма-сервер – це ...	<p>a) програма, яка забезпечує надання відповідного набору послуг з боку сервера;</p> <p>b) підпрограма, яка забезпечує надання відповідного набору послуг з боку сервера;</p> <p>c) програма, яка аналізує надання відповідного набору послуг з боку сервера;</p> <p>d) програма, яка припиняє надання відповідного набору послуг з боку сервера.</p>
57	Програма-клієнт – це ...	<p>a) програма, яка забезпечує доступ до набору послуг сервера з боку клієнта;</p> <p>b) програма, яка обслуговує сервер;</p> <p>c) програма, яка містить послуги сервера з боку клієнта;</p> <p>d) програма, яка аналізує набір послуг сервера з боку клієнта.</p>
58	У програмі «клієнт-сервер» можна виділити такі логічні компоненти:	<p>a) 1) компонент представлення (подання) (presentation), 2) прикладний компонент (business application), 3) компонент доступу до інформаційних ресурсів (resource manager);</p> <p>b) 1) компонент аналізу (analyze), 2) прикладний компонент (business application), 3) компонент доступу до інформаційних ресурсів (resource manager);</p> <p>c) 1) компонент представлення (подання) (presentation), 2) прикладний компонент (business application), 3) компонент обчислень (calculation);</p> <p>d) 1) компонент представлення (подання) (presentation), 2) технічний компонент (technique application), 3) компонент доступу до інформаційних ресурсів (resource manager).</p>
59	Функції інформаційного забезпечення:	<p>a) 1) контроль за діяльністю підприємства або галузі, для якого створена інформаційна система, 2) підготовка для обробки первинних даних, що відображають стан керованих об'єктів, 3) забезпечення автоматизованої обробки даних, 4) здійснення прямого і зворотного зв'язків між об'єктами і суб'єктами управління;</p> <p>b) 1) контроль за діяльністю підприємства або галузі, для якого створена інформаційна система, 2) підготовка для обробки первинних даних, що відображають стан керованих об'єктів, 3) забезпечення автоматизованої обробки даних;</p> <p>c) 1) контроль за діяльністю підприємства або галузі, для якого створена інформаційна система, 2) підготовка для обробки оброблених даних, що відображають стан керованих об'єктів, 3) забезпечення ручної обробки даних, 4) здійснення прямого і зворотного зв'язків між об'єктами і суб'єктами управління;</p> <p>d) 1) контроль за діяльністю підприємства або галузі, для якого створена інформаційна система, 2) підготовка для обробки первинних даних, що відображають стан керованих об'єктів, 3) забезпечення автоматизованої обробки даних, 4) здійснення впливу на об'єкти і суб'єкти управління.</p>

60	Позамашинне інформаційне забезпечення охоплює:	<p>a) 1) систему класифікації і кодування інформації, 2) систему документування, 3) схему інформаційних потоків і документообігу;</p> <p>b) 1) систему класифікації і кодування інформації, 2) систему проектування, 3) систему інформаційних потоків і документообігу;</p> <p>c) 1) систему перекодування інформації, 2) систему документування, 3) схему потоків і документообігу;</p> <p>d) 1) систему класифікації і кодування інформації, 2) схему інформаційних потоків і документообігу.</p>
61	Класифікація – це ...	<p>a) система розподілу об'єктів за типами або класами згідно з вибраними ознаками;</p> <p>b) система розподілу об'єктів за ознаками згідно з вибраними класами;</p> <p>c) система перерозподілу даних за типами або класами згідно з вибраними ознаками;</p> <p>d) система аналізу об'єктів згідно з вибраними ознаками.</p>
62	Вимоги до класифікації:	<p>a) 1) повнота системи класифікації, 2) однозначність реквізитів (атрибутів), 3) можливість внесення в класифікацію нових об'єктів (можливість розширення класу);</p> <p>b) 1) повнота системи класифікації, 2) однозначність реквізитів (атрибутів);</p> <p>c) 1) повнота системи класифікації, 2) неоднозначність реквізитів (атрибутів), 3) можливість внесення в класифікацію нових об'єктів (можливість розширення класу);</p> <p>d) 1) повнота системи класифікації, 2) однозначність реквізитів (атрибутів), 3) неможливість внесення в класифікацію нових об'єктів (можливість розширення класу).</p>
63	Види класифікаторів:	<p>a) міжнародні, загальнодержавні, галузеві, локальні;</p> <p>b) ієрархічні, фасетні;</p> <p>c) порядкові, серійні;</p> <p>d) класифікаційні, реєстраційні.</p>
64	Системи класифікації:	<p>a) ієрархічна, фасетна;</p> <p>b) міжнародна, загальнодержавна, галузева, локальна;</p> <p>c) порядкова, серійна;</p> <p>d) класифікаційна, реєстраційна.</p>
65	Системи кодування:	<p>a) класифікаційна, реєстраційна;</p> <p>b) порядкова, серійна;</p> <p>c) ієрархічна, фасетна;</p> <p>d) порозрядна, повторення, комбінована.</p>
66	Методи класифікаційного кодування:	<p>a) порозрядний, повторення, комбінований;</p> <p>b) порядковий, серійний;</p> <p>c) класифікаційний, реєстраційний;</p> <p>d) ієрархічний, фасетний.</p>
67	Методи реєстраційного кодування:	<p>a) порядковий, серійний;</p> <p>b) порозрядний, повторення, комбінований;</p> <p>c) класифікаційний, реєстраційний;</p> <p>d) ієрархічний, фасетний.</p>

68	Внутрішньомашинне інформаційне забезпечення охоплює ...	<ul style="list-style-type: none"> a) масиви і бази даних на машинних носіях та програми доступу до цих даних; b) літературу і документацію до сфери використання; c) файли даних на машинних носіях; d) засоби доступу до баз даних.
69	При формуванні електронного цифрового підпису використовуються ...	<ul style="list-style-type: none"> a) взаємопов'язані компоненти: закритий (або секретний) ключ; відкритий ключ; зміст документа; ідентифікатор відправника документа; b) незалежні компоненти: закритий (або секретний) ключ; відкритий ключ; зміст документа; ідентифікатор відправника документа; c) взаємопов'язані компоненти: закритий (або секретний) ключ; зміст документа; ідентифікатор відправника документа; d) взаємопов'язані компоненти: закритий (або секретний) ключ; відкритий ключ.
70	Таблиці MS Access призначені для ...	<ul style="list-style-type: none"> a) збереження даних бази даних і виконання операцій фільтрування та сортування даних; b) автоматизації часто повторюваних функцій обробки даних; c) подання даних у зручному вигляді на екрані монітора та для управління базою даних; d) вибору та обробки даних.
71	Запити MS Access призначені для ...	<ul style="list-style-type: none"> a) вибору та обробки даних, що містяться в таблицях та інших запитах; b) автоматизації часто повторюваних функцій обробки даних; c) подання даних у зручному вигляді на екрані монітора та для управління базою даних; d) вибору та обробки даних.
72	Форми MS Access призначені для ...	<ul style="list-style-type: none"> a) подання даних у зручному вигляді на екрані монітора та для управління базою даних; b) автоматизації часто повторюваних функцій обробки даних; c) подання бази даних у зручному вигляді на екрані монітора та для управління базою даних; d) вибору та обробки даних.
73	Звіти MS Access призначені для ...	<ul style="list-style-type: none"> a) подання даних у друкованому вигляді; b) автоматизації часто повторюваних функцій обробки даних; c) подання даних у зручному вигляді на екрані монітора та для управління базою даних; d) вибору та обробки даних.
74	Макроси MS Access призначені для ...	<ul style="list-style-type: none"> a) автоматизації часто повторюваних функцій обробки даних; b) подання даних у друкованому вигляді; c) подання даних у зручному вигляді на екрані монітора та для управління базою даних; d) реєстрації часто повторюваних функцій обробки даних.

75	Серед інформаційних задач з позиції правил організації вирішення задач у встановленому управлінському режимі розрізняють:	<ul style="list-style-type: none"> a) регламентні і запитно-довідкові; b) систематичні, епізодичні, випадкові; c) термінові і нетермінові; d) типові і нетипові.
76	Серед інформаційних задач з регулярності вирішення розрізняють:	<ul style="list-style-type: none"> a) систематичні, епізодичні, випадкові; b) термінові і нетермінові; c) типові і нетипові; d) одиничні, групові, масові.
77	Серед інформаційних задач за терміном вирішення розрізняють:	<ul style="list-style-type: none"> a) термінові і нетермінові; b) систематичні, епізодичні, випадкові; c) типові і нетипові; d) одиничні, групові, масові.
78	Серед інформаційних задач з позиції технологічної спільності правил вирішення задач розрізняють:	<ul style="list-style-type: none"> a) типові і нетипові; b) термінові і нетермінові; c) систематичні, епізодичні, випадкові; d) одиничні, групові, масові.
79	Серед інформаційних задач за повторюваністю розрізняють:	<ul style="list-style-type: none"> a) одиничні, групові, масові; b) типові і нетипові; c) термінові і нетермінові; d) систематичні, епізодичні, випадкові.
80	Серед інформаційних задач за математичною основою обчислень розрізняють:	<ul style="list-style-type: none"> a) задачі прямого розрахунку, оптимізаційні задачі, багатоваріантні задачі з пошуком найкращих рішень; b) типові і нетипові; c) термінові і нетермінові; d) систематичні, епізодичні, випадкові.
81	Серед інформаційних задач з позиції правил організації вирішення задач у встановленому управлінському режимі розрізняють:	<ul style="list-style-type: none"> a) регламентні і запитно-довідкові; b) систематичні, епізодичні, випадкові; c) термінові і нетермінові; d) типові і нетипові.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 1. Організація баз даних і знань : підручник. Львів : «Магнолія-2006», 2020. 440 с.
2. Берко А. Ю., Верес О. М., Пасічник В. В. Системи баз даних та знань. Книга 2. Системи управління базами даних та знань : навчальний посібник (рек. МОН України). Львів : «Магнолія-2006», 2021. 584 с.
3. Гайдаржи В. І., Ізварін І. В. Бази даних в інформаційних системах. Київ : Ун-т «Україна», 2018. 417 с.
4. Гайдаржи В. І., Дацюк О. А. Основи проектування та використання баз даних. К. : Політехніка, 2004. 256 с.
5. Гарсиа-Молина Г., Ульман Дж., Уидом Дж. Системы баз данных. Полный курс. М. : «Вильямс», 2003. 1088 с.
6. Грофф Дж. Р. SQL. Полное руководство. СПб. : «Вильямс», 2015. 959 с.
7. Дейт К. Дж. Введение в системы баз данных ; 8-е издание. М. : «Вильямс», 2005. 1328 с.
8. Джексон Г. Проектирование реляционных баз данных для использования с микроЭВМ. М. : Мир, 1991. 252 с.
9. Завадський І. О. Основи баз даних : навч. посіб. К. : Видавець І. О. Завадський, 2011. 192 с.
10. Зарицька О. Л. Бази даних та інформаційні системи : метод. посіб. Житомир : Вид-во ЖДУ ім. І. Франка, 2009. 132 с.
11. Ковальчук А. М., Левицький В. Г. Принципи проектування баз даних : навчальний посібник. Ж. : ЖДТУ, 2009. 123 с.
12. Коннолли Т., Бег Л. Базы данных. Проектирование, реализация и сопровождение. Теория и практика ; 3-е издание. М. : «Вильямс», 2003. 1440 с.
13. Крёнке Д. Теория и практика построения баз данных ; 8-е изд. СПб. : Питер, 2003. 800 с.: ил. (Серия «Классика computer science»).
14. Пасічник В. В., Резніченко В. А. Організація баз даних і знань. Київ : Видавнича група ВНУ, 2006. 384 с.
15. Пасічник В. В., Резніченко В. А. Організація баз даних та знань. К.: Видавнича група ВНУ, 2006. 384 с.
16. Петух А. М. Романюк О. Н. Романюк О. В. Бази даних. Мови запитів, управління транзакціями, розподілена обробка даних [Електронний ресурс] 2016. Режим доступу до ресурсу : https://web.posibnyku.vntu.edu.ua/fitki/11petuh_bazdanyh_movy_zalitiv/zmist.htm.
17. Райордан Р. Основы реляционных баз данных ; пер, с англ. М. : Издательско-торговый дом «Русская Редакция», 2001. 384 с. : ил.

18. Роб П. Системы баз данных: проектирование, реализация и управление. СПб. : БХВ-Петербург, 2004. 1040 с.
19. Романюк О. Н., Романюк О. В., Денисюк А. В. Методичні вказівки до виконання курсової роботи з дисципліни «Бази даних» для студентів спеціальності 121 «Інженерія програмного забезпечення» денної та заочної форм навчання. Вінниця : ВНТУ, 2018. 52 с.
20. Романюк О. Н., Савчук Т. О. Організація баз даних і знань .Вінниця : УНІВЕРСУМ-Вінниця, 2003. 217 с.
21. Тарасов О. В., Федько В. В., Лосев М. Ю. Проектування баз даних : навч. посібн. Х. : Вид. ХНЕУ, 2011. 200 с
22. Тиори Т., Фрай Дж. Проектирование структур баз данных. В 2 кн., М. : Мир, 1985. Кн. 1. 287 с.: Кн. 2. 320 с.
23. Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данняху: учебник для высших учебных заведений СПб. : КОРОНА принт, 2004. 736 с.
24. Цикритизис Д., Лоховски Ф. Модели данных. М. : Финансы и статистика, 1985. 344 с.
25. Четвериков В. Н., Ревунков Г. И., Самохвалов Э. Н. Базы и банки данных. М.: Высшая школа, 1993.
26. Шаров С. В., Осадчий В. В. Бази даних та інформаційні системи : навчальний посібник. Мелітополь : Вид-во МДПУ ім. Б. Хмельницького, 2014. 352 с.

*Електронне навчальне видання
комбінованого використання.
Можна використовувати в локальному та мережному режимах*

**Романюк Олександр Никифорович,
Романюк Оксана Володимирівна,
Денисюк Алла Василівна**

Практикум для самостійної роботи студентів з навчальної дисципліни «Бази даних»

Практикум

Рукопис оформлено *О. Романюком*

Редактор *В. Дружиніна*

Оригінал-макет виготовлено *О. Кушнір*

Підписано до видання 10.08.2022 р.
Гарнітура Times New Roman.
Зам. № P2022-064.

Видавець та виготовлювач
Вінницький національний технічний університет,
Редакційно-видавничий відділ.
ВНТУ, ГНК, к. 114.
Хмельницьке шосе, 95, м. Вінниця, 21021.
Тел. (0432) 65-18-06.
press.vntu.edu.ua;
Email: irvc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009