

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
Факультет інформатики та обчислювальної техніки

Кафедра комп'ютерних
систем та мереж

МЕТОДИЧНІ ВКАЗІВКИ
до виконання контрольної роботи
з дисципліни “Організація баз даних”
для студентів спеціальності 123 “Комп'ютерна інженерія”
заочної форм навчання

2019

Методичні вказівки до виконання контрольної роботи з дисципліни “Організація баз даних” для студентів спеціальності 123 “Комп’ютерна інженерія” заочної форми навчання /Укл.: Т.О. Паромова, - Запоріжжя: НУ ”Запорізька політехніка”, 2019 – 36 с.

Укладачі: Т.О.Паромова, ст.викладач.

Рецензент: С.Д. Точилін, доцент, к.ф.-м.н.

Відповідальний за випуск: Паромова Т.О., ст.викладач.

Затверджено
на засіданні кафедри
“Комп’ютерні системи та мережі”
Протокол №4 от 15.11.2019

Рекомендовано до видання
НМК факультету комп’ютерних наук
і технологій
Протокол №3 от 18.11.2019.

ЗМІСТ

1. РЕКОМЕНДАЦІЇ ЩОДО ВИКОНАННЯ ТА ОФОРМЛЕННЯ КОНТРОЛЬНОЇ РОБОТИ.....	4
2. КОНТРОЛЬНІ ЗАВДАННЯ	6
2.1 Завдання 1 Операції реляційної алгебри	6
2.2 Нормалізація відношень	12
2.3 Створення бази даних	21
2.4 Створення SQL- запитів.....	23
2.5 КОНТРОЛЬНІ ПИТАННЯ	32
3. РЕКОМЕНДОВАНА ЛІТЕРАТУРА	34

1. РЕКОМЕНДАЦІЇ ЩОДО ВИКОНАННЯ ТА ОФОРМЛЕННЯ КОНТРОЛЬНОЇ РОБОТИ

Завдання до контрольної роботи складаються з трьох теоретичних питань та практичного завдання. Структури кожного завдання подібні.

Контрольна робота виконується кожним студентом за варіантом, який визначає викладач. Відповідні питання всіх варіантів мають однакову складність.

До виконання контрольної роботи слід приступати після вивчення теоретичного матеріалу, який надається у спеціальній літературі згідно переліку. При цьому не слід обмежуватись запропонованим переліком літератури, а рекомендується скористатись додатковими джерелами інформації, вивчити всебічно задані питання, узагальнити відомості з вивчених питань.

При оформленні домашньої контрольної роботи пропонується спочатку записувати питання, а потім скласти відповідь. Відповіді на питання мають бути конкретними але повними, при цьому не припустимо використовувати скорочення слів. При необхідності відповідь може містити графіки, малюнки, таблиці. Відповідь на кожне питання має закінчуватись посиланням на літературу, яка використовувалась при опрацюванні питання. Результати виконання кожного завдання контрольної роботи необхідно надрукувати на окремому аркуші паперу з обов'язковим зазначенням номеру завдання. Для показу результатів виконання завдання 2 необхідно надрукувати:

- структуру бази даних;
- структуру та результати виконання запитів;
- звіти;
- текст та результати виконання SQL запиту.

На кожному листі треба надрукувати текст завдання, яке пояснює результат виконання відповідного завдання.

Контрольна робота оформлюється на аркушах паперу формату А4 (210х297 мм) або у тонких зошитах. Робота може бути виконана рукописним способом або за допомогою комп'ютеру на одному боці аркуша білого паперу. При використанні принтеру текст роботи друкується шрифтом типу Time New Roman, розміром 12. При рукописному способі оформлення роботи текст має бути написаний розбірли-

ве та охайно. При оформленні тексту контрольної роботи слід дотримуватись таких розмірів берегів: верхній, нижній, лівий – не менш 20 мм, правий - не менш 10 мм.

Всі листи контрольної роботи повинні бути пронумеровані. Нумерація сторінок встановлюється посередині верхнього поля документу. Титульний лист не нумерується. Примірник оформлення титульного листа домашньої контрольної роботи надається у додатку А.

Аркуші контрольної роботи обов'язково з'єднуються будь яким способом.

Виконана контрольної роботи складається з титульного листа, відповідей на питання згідно варіанту, друковані результати виконання кожного практичного завдання, переліку літератури. Виконані практичні завдання надаються на дискеті. Пояснення, що до їх виконання, студент надає при захисті контрольної роботи.

2. КОНТРОЛЬНІ ЗАВДАННЯ

2.1 Завдання 1 Операції реляційної алгебри

Для виконання першого завдання контрольної роботи необхідні знання з основ реляційної алгебри. При оформленні завдання №1 необхідно послідовно показати відношення зі змінами після виконання кожної операції реляційної алгебри.

Основні операції реляційної алгебри

Вибірка дає в результаті відношення, що містить усі кортежі, які задовольняють деякій умові $\sigma_{\text{предикат}}(\mathbf{R})$

де предикат – умови відбору кортежів з відношення $\mathbf{R}$, умови відбору кортежів можуть містити оператори порівняння ($=, <, >$).

Приклад

Відношення $\mathbf{R}$

A	B	C	D	E
1	z	y	1	1
2	zz	yy	2	2
3	zz	yy	6	7
3	zz	y	4	2

$\sigma_{a=3 \text{ and } d>2}(\mathbf{R})$

A	B	C	D	E
3	zz	yy	6	7
3	zz	y	4	2

Проекція - відношення, що містить тільки деякі атрибути відношення $\mathbf{R}$, після витягу зазначених атрибутів і виключення з результату рядків – дублікатів.

$\pi_{\text{атр1}, \text{атр2}}(\mathbf{R})$

де атр1 , атр2 – атрибути відношення, які мають бути перенесені у результат проекції

Приклад

Відношення R

A	B	C	D	E
1	z	y	1	1
2	zz	yy	2	2
3	zz	yy	6	7
3	zz	y	4	2

 $\Pi_{A,E}(R)$

A	E
1	1
2	2
3	7
3	2

Декартовий добуток – відношення, усі кортежі якого є сполученням кортежів двох різних відносин

Позначення $R1 * R2$

Приклад

R1

A
1
2

R2

C
y
yy

R1 * R2

A	C
1	y
1	yy
2	y
2	yy

Віднімання повертає відношення, усі кортежі якого належать тільки першому з двох відносин R і S.

Позначення $R - S$

Приклад

R

A	B	C
x1	κ	1
x2	κ1	1

S

A	B	C
x2	κ1	1
x1	d1	5
x1	κ1	4

R - S

A	B	C
x1	κ	1

Об'єднання двох сумісних по типі відношень повертає відношення з тим же заголовком і тілом із усіма кортежами, що належать відношенню A чи B або обом відношенням одночасно. Кожен елемент включається в результуюче відношення тільки раз.

Позначення $R \cup S$

Приклад

R

A	B	C
x1	к	1
x2	к1	1

S

A	B	C
кк	d1	4
к	d1	5

$R \cup S$

A	B	C
x1	к	1
x2	к1	1

Перетинання повертає відношення, усі кортежі якого належать одночасно двом відношенням R і S

Позначення $R \cap S = R - (R - S)$

Приклад

R

A	B	C
x1	к	1
x2	к1	1

S

A	B	C
x2	к1	1
x1	d1	5
x1	к1	4

$R \cap S$

A	B	C
x2	к1	1

Відношення, над якими здійснюються операції об'єднання та перетинання, повинні мати схеми з ідентичним набором атрибутів, що подаються в одному порядку (однакові заголовки) і всі атрибути базуються на співпадаючих доменах.

Природне з'єднання передбачає включення в підсумкове відношення тих кортежів відношень R і S , що збігаються в атрибутах, загальних для схем R і S .

Позначення $R \bowtie S$.

Приклад

R

A	B	C
x1	к	1
x2	к1	1

S

A	D	E
x2	к1	1
x1	d1	5
x1	к1	4

R▷<**S**

A	B	C	D	E
x1	κ	1	d1	5
x1	κ	1	κ1	4
x2	κ1	1	κ1	1

R

A	B	C
x1	κ	1
x2	κ1	1

S

A	D	E
x2	κ1	1
x1	d1	5
x1	κ1	4

R▷<**S**

A	B	C	D	E
x1	κ	1	d1	5
x1	κ	1	κ1	4
x2	κ1	1	κ1	1

Зовнішнє з'єднання двох таблиць у разі необхідності збереження всіх значень одного з атрибутів при відсутності відповідного значення у іншій таблиці використовують зовнішнє з'єднання.

Ліве зовнішнє з'єднання **R**▷<**S** – всі кортежі відношення **R**, для яких не існує відповідних значень у відношенні **S**, також включаються у відношення **S**. Для відсутніх значень атрибутів відношення **S** у результуюче відношення пропонується значення **NULL**.

Праве відкрите з'єднання створюється таким самим чином.

2.1.1 Варіанти завдань з теми “Операції реляційної алгебри”

При необхідності спростити вираз та знайти результат

Варіант 1.

R1

A	B	C
F	1	X
G	1	Y
	3	Z
H	8	X

R2

B	D	E
1	KK	S
8	LL	T
7	LL	T

R3

E	D
T	8
S	1

$$F = \Pi_{D,E}(R1 \triangleright \triangleleft R2) \cup R3$$

Вариант 2.**R1**

A	B	C
x1	κ	1
x1	κκ	1
x2	κ	2
x3	κκ	3

R2

B	D	E
κκ	d1	4
κ	d1	5
κ	d2	5

R3

E	F
5	f
6	ff
7	ff

$$F = \sigma_{E=5} (\sigma_{B=\kappa\kappa} (\Pi_{B,C}(R1) * (\Pi_E(R2) \cup \Pi_E(R3))))$$

Вариант 3.**R**

A	B	C	D	E
1	Z	3	1	7
2	Zz	5	2	2
3	Z	3	6	7
3	Z	3	4	2

$$F = \Pi_{BD}(\sigma_{E=7}(\sigma_{D \geq 3}(R) \cap \sigma_{B=Z}(R)))$$

Вариант 4.**R**

A	B	C	D	E
1	z	y	1	1
2	zz	yy	2	2
3	zz	yy	6	7
3	zz	y	4	2

$$F = \sigma_{A=3}(\Pi_{B,D}(\sigma_{D < 6}(R) \triangleright \triangleleft (\Pi_{A,B}(\sigma_{B=ZZ}(R))))$$

Вариант 5.

R				
A	B	C	D	E
1	z	y	6	1
2	zz	yy	2	2
3	zz	yy	6	7
3	zz	y	4	2

$$F = \sigma_{A < 3}(\Pi_B(\sigma_{D=6}(R))) \triangleright \triangleleft (\Pi_{A,B,C}(\sigma_{B=ZZ \text{ And } C=YY}(R)))$$

Вариант 6.

R1		
A	B	C
x1	κ	1
x1	κκ	1
x2	κ	2
x4	κκ	3

R2		
B	D	E
κκ	d1	4
κκ	d1	5
κ	d2	5

R3	
E	F
5	f
6	ff
7	ff

$$F = \sigma_{D=d2}(\Pi_{B,D}(R1 \triangleright \triangleleft R2)) \cap (\Pi_{B,D}(R2 \triangleright \triangleleft R3))$$

Вариант 7.

R1		
A	B	C
x1	κ	1
x1	κκ	1
x2	κ	2
x4	κκ	3

R2		
B	D	E
κκ	d1	4
κκ	d1	5
κ	d2	5

R3	
E	F
5	f
6	ff
7	ff

$$F = (\sigma_{R1.B=\kappa}(\Pi_{b,c}(R1))) \triangleright \triangleleft (\Pi_{B,E}(R2) \triangleright \triangleleft (\Pi_E(R3)))$$

Варіант 8.

A	B	C		D	E
1	Z	3		1	1
2	Zz	5		2	2
3	Z	3		6	7
3	Z	3		4	2

$$F = \sigma_{B=Z} (\Pi_{A,B}(R)) \supset \sigma_{E=2} (\Pi_{B,E}(R)) \supset \sigma_{C=3} (\Pi_{C,E}(R))$$

Варіант 9.**R1**

A	B	C
x1	к	1
x1	кк	1
x2	кк	2

R2

B	D	E
кк	d1	4
кк	d1	5
к	d2	5

R3

E	F
5	f
5	ff
7	ff

$$F = \sigma_{R1.B=кк} (\Pi_{B,C}(R1)) \supset \Pi_{B,E}(R2) \supset \sigma_{E=5}(R3)$$

2.2 Нормалізація відношень

Для виконання 2 завдання необхідно засвоїти матеріал з теми „Реляційна модель баз даних” та „Нормалізація таблиць баз даних”.

В реляційній СУБД користувач сприймає базу даних, як сукупність відношень.

Відношення – це двомірна таблиця, яка складається зі стовпців та рядків. У кожному відношенні зберігається інформація про якусь сутність.

Сутність – це будь-який окремий предмет, інформацію про який треба зберігати в БД. Тип сутності являє собою множину об'єктів реального світу з однаковими властивостями. Кожен тип сутностей ідентифікується іменем та списком властивостей. Найменша одиниця даних реляційної моделі - це атомарне значення даних, неподільне для даної моделі.

Елементами відношень є кортежі. Кортеж – це рядок у відношенні. Атрибут – поименований стовпчик відношення. Кожен атрибут має своє ім'я. Один або декілька атрибутів відношення призначаються ключовими.

Ключ - це мінімальний набір атрибутів, який унікально ідентифікує окремі екземпляри сутностей, яким призначене відношення. Мінімальність означає, що виключення з ключового набору атрибутів будь якого атрибуту не дозволить ідентифікувати сутність за останніми атрибутами. У будь який момент часу не існує двох кортежів з однаковими значеннями відповідних атрибутів.

Потенційний ключ-атрибут або набір атрибутів, які однозначно ідентифікують окремі екземпляри сутностей.

Первинний ключ – унікальний ідентифікатор таблиці, тобто атрибут або така комбінація атрибутів, які однозначно ідентифікують значення атрибутів одного кортежу.

Зв'язок – асоціювання двох або більше сутностей. На практиці частіше використовуються зв'язок типів 1:1(один до одного), 1:M(один до багатьох), M:M(багато до багатьох).

Функціональна залежність описує зв'язок між атрибутами відношення. Якщо в відношенні R атрибут B функціонально залежить від атрибуту A, то кожне значення атрибуту A зв'язано тільки з одним значенням атрибуту B. Це позначається як $A \rightarrow B$. Таким чином, у разі існування функціональної залежності між атрибутами A та B в кожному рядку, які містять однакове значення атрибуту A, буде міститися однакове значення атрибуту B.

Якщо всі з не ключові атрибути повністю залежать від всіх ключових, то функціональна залежність $A \rightarrow B$ між атрибутами A та B є певною, як що при вилученні якого не будь атрибуту з A, залежність повністю зникає.

Частковою функціональною залежністю називається залежність, якщо де який не ключовий атрибут залежить тільки від частки ключових атрибутів.

Залежність між атрибутами A та B називається транзитивною, як що існують залежності $A \rightarrow B$ та $B \rightarrow C$, тоді залежність $A \rightarrow C$ є транзитивною.

Нормалізація – це процес декомпозиції таблиць з метою усунення аномалій оновлення даних та надмірності інформації у БД. Нормалізовані таблиці відповідають вимогам нормальних форм.

Таблиця знаходиться у 1НФ, якщо на перетині кожного рядку та кожного стовпця знаходиться тільки одне значення відповідного атрибуту.

Таблиця відповідає 2НФ, якщо вона відповідає 1НФ та кожен атрибут цієї таблиці, якій не входить до складу первинного ключа, характеризується певною функціональною залежністю від первинного ключа.

Таблиця відповідає 3НФ, якщо вона відповідає 1НФ та 2НФ та не має атрибутів, які не входять в первинний ключ та знаходяться у транзитивні залежності від первинного ключа.

Послідовність проведення нормалізації бази даних.

Предметна область – деканат

Розв’язувані задачі: облік успішності студентів (Прізвище, ім’я та по батькові, дисципліни, що вивчаються, оцінка), Оцінка визначається за 5 бальною системою, у разі відсутності студента на іспиті у графі Оцінка остається пуста комірка

На підставі аналізу предметної області таблиця Успішність матиме вигляд:.

Успішність(КодСтудента, Студент, Дисципліни, Оцінка);

- Для приведення таблиці Успішність до 1 нормальної форми пусті комірки заповнюються відповідною інформацією.
- Визначається повторювальна група відомостей, яка повторюється у різних студентів

Повторювальна група = (дисципліни, оцінка)

Наявність повторювальної групи призводить до появи декількох значень атрибутів Дисципліни та Оцінка для одного значення ключа Код_Студента

Успішність

Код Студента	Студент	Дисципліни	Оцінка
01	Сич Д.П.	Основи екології	4
		Програмування	4
		Вища математика	3
02	Кім Ф.В.	Основи екології	5
		Програмування	5
		Фізика	5
		Вища математика	5
03	Куль А.О.	Основи екології	3
		Програмування	5
		Фізика	
		Вища математика	5

(Таблиця є прикладом, та приводить відповідну таблицю в контрольній роботі не потрібно)

- Для приведення таблиці Успішність до 1 нормальної форми пусті комірки заповнюються відповідною інформацією.

- Визначається повторювальна група відомостей, яка повторюється у різних студентів

Повторювальна група = (дисципліни, оцінка)

Наявність повторювальної групи призводить до появи декількох значень атрибутів Дисципліни та Оцінка для одного значення ключа Код_Студента

- Повторювальна група усувається шляхом заповнення порожніх комірок відповідними записами. Далі визначаються атрибути, в яких можуть повторюватися значення ТЕКСТОВОЇ інформації, тому що данні саме цього типу займають найбільше місці на жорсткому диску та призводять до появи різних аномалій при обробці інформації в базі даних. Зайва інформація у таблиці Успішність студентів може призвести до аномалій оновлення(оновлення даних, вилучення та введення нових даних)

- У таблиці Успішність текстові дані можуть повторюватись у стовпцях Студент та Дисципліни. Тому ці стовпці треба винести у окремі таблиці, а в таблиці Успішність залишити копії ключів.

- Створюються нові таблиці

Студент

Студент(Код_Студента, Студент),
ключ - Код_Студента

Код Студента	Студент
01	Сич Д.П.
02	Кім Ф.В.
03	Куль А.О.

Дисципліни
Дисципліні(Код_Дисципліни, Дисципліна)
ключ - Код_Дисципліни

код_Дисципліни	Дисципліна
01	Основи екології
02	Програмування
03	Фізика
04	Вища математика

А у таблиці Успішність замість текстових значень вказуються ключи відповідної таблиці. Наприклад, замість студента Кім Ф.В. в таблицю Успішність вноситься значення ключа 02, а замість дисципліни Фізика в таблицю Успішність вноситься значення ключа 03
Тоді таблиця Успішність має вигляд:

Успішність(КодСтудента, , Код_Дисципліни, Оцінка);

Як можна бачити модифікована таблиця Успішність містить тільки числові дані, як результат займає менше місця на диску, та усуваються можливі аномалії обробки даних.

Таким чином, таблиця Успішність відповідає 1 НФ.

- Далі для приведення таблиці до відповідності 2 НФ необхідно вибрати первинний ключ у таблиці та визначити функціональні залежності між Успішність атрибутами відношення. Часткова функціональна залежність, яка усувається у 2 НФ, також може призвести до появи аномалій.

У прикладі, що розглядається, однозначно атрибути КодСтудента та КодДисц визначають всі записи у атрибуті Оцінка, Вони обираються складним первинним ключем. Кожен атрибут по одинці не

може однозначно визначити значення атрибуту Оцінка, тому таблиця Успішність відповідає 2 НФ.

Успішність

Код Студента	Код_Дисципліни	Оцінка
01	01	4
01	02	4
01	03	5
01	04	3
02	01	5
02	02	5
02	03	5
02	04	5
03	01	3
03	02	5
03	03	
03	04	5

Якщо проаналізувати вихідну таблицю Успішність, то можна визначити наступні функціональні залежності

Функціональні залежності вихідної таблиці Успішність

Детермінант	Залежні атрибути	Вид залежності
КодСтудента, КодДисц	Оцінка	первинний ключ
КодСтудента	Студент	часткова залежність $\Rightarrow$ залежність не від обох атрибутів ключа, а тільки від атрибуту КодСтудента
КодДисципліни	Дисципліна	часткова залежність $\Rightarrow$ залежність не від обох атрибутів ключа, а тільки від КодДисципліни

1. У таблиці Успішність визначено дві часткові функціональні залежності, тому необхідно створити нові таблиці, куди переміс-

тити відповідні атрибути з копією частки первинного ключа, від якої вони залежать. Таблицю Успішність розділяється на три таблиці

Таблиця Студенти

КодСтудента	ПІБ
01	Сич Д.П.
02	Кім Ф.В
03	Куль А.О.

Таблиця Дисципліни

Код-Дисц	дисципліни
01	Основи екології
02	Програмування
03	Фізика
04	Вища математика

Тоді таблиця Успішність складається з зовнішніх ключів та атрибуту Оцінка. Але саме цей результат був отриманий на першому етапі проектування. Аналіз відповідності отриманої таблиці Успішність на відповідність 2 НФ є перевірка правильності проектування

Таблиця Успішність

Код Студента	Код_Дисципліни	Оцінка
01	01	4
01	02	4
01	03	5
01	04	3
02	01	5
02	02	5
02	03	5
02	04	5
03	01	3
03	02	5
03	03	
03	04	5

Всі три таблиці відповідають 2НФ. Створені таблиці мають схеми:
 Дисципліни (КодДисц, Дисципліна)
 Студенти (КодСтудента, ПІБ)
 Успішність (КодСтудента, КодДисц, Оцінка)

- Наступний етап нормалізації – приведення таблиці до відповідності 3 НФ – тобто у таблиці не повинно бути транзитивне залежних атрибутів, тобто неключові атрибути не повинні мати функціональну залежність між собою. Для відповідності таблиці Успішність в 3НФ треба усунути транзитивну залежність, Але у попередньому аналізі транзитивна залежність не була визначена, то модифікована таблиця Успішність відповідає 3 НФ

- Наступний етап нормалізації - перевірка відповідності створених таблиць НФБК (нормальна форма Бойса-Кодда)

- Тому що всі детермінанти є потенційними ключами всі таблиці знаходяться у НФБК.

- Таким чином створювана БД містить наступні таблиці
Дисципліни (КодДисц, Дисципліна);
Студенти (КодСтудента, Студент);
Успішність (КодСтудента, КодДисц, Оцінки);

2.2.1 Варіанти завдань "Нормалізація відношень"

Нормалізувати базове відношення та визначити обмеження цілісності.

Варіант 1. Розклад

Викладач	День тижня	Пара	Дисципліна	Група

Варіант 2. Аптека

Група ліків	Гіки	Придатний до	Ціна	Кількість	Аптека	Господарник

Варіант 3. Передплата

Код видання	Видання	Вид видання	Ціна на рік	Дата замовлення	Термін замовлення	Замовник	Адреса замовника

Варіант 4. Ремонт квартир

Дата	Рег №	Замов-ник	Адреса замовника	Вид робіт	Вартість роботи	Таб № виконавця	Викона-вця	Спеці-альність

Варіант 5. Реєстрація документів

№ документу	Дата надхо-дження	Вид документу	Документу	Відділ	Викона-вць	Дата виконання
1	01.02.19	закон	Закон про	2	Бех І.І	14.02.19

Варіант 6. Страхова компанія

№	Дата	Філія	Адреса філії	Телефон філії	Вид страхування	Тарифна ставка страхування	Клієнт	Агент

Варіант 7. Рецепти

Код блюда	Блюдо	Продукти	Кількість на 1 порцію	Ціна продук-ту за кг	Постачаль-ник

Варіант 8. Рейтинг пісень

№ п/п	пісня	виконавець	тривалість	компози- тор	поет	рейтинг

Варіант 9. Відділ кадрів

Таб.№	Дата прийому	Працівник	Дата народження	Освіта	Діти	Дата народ- дження дітей	Дом телефон	відділ	Посада	Оклад

2.3 Створення бази даних**2.3.1 Варіанти завдань “Створення бази даних**

У середовищі СУБД за результатами виконання завдання 2:

- створити таблиці та встановити зв'язок між ними
- створити форми для заповнення всіх таблиць
- заповнити таблиці 10 записами
- за допомогою запитів визначити зазначену інформацію
- створити звіти за результатами запитів

Варіант 1. Розклад

- Зайнятість викладача у тиждень (кількості занять *2)
- Які викладачі працюють у групи, номер якої вводиться з клавіатури, у понеділок
- Які дисципліни читає викладач у зазначений день тижня

Варіант 2. Аптека

- У якій аптеці наявний необхідний препарат (найменування ліків вводиться з клавіатури)
- Перелік ліків, для яких закінчився термін використання
- Суму, на яку наявні ліки у всіх аптеках, згрупувати по кожному препарату окремо

Варіант 3. Передплата

- Які видання замовлені на рік на поточну дату (назва видання вводиться з клавіатури)
- Які видання замовлені за останні 10 днів
- Список замовників з зазначенням адреси, назви видання та терміну замовлення

-

Варіант 4. Ремонт квартир

- Які роботи та яким виконавцем виконуються на поточну дату
- Скільки замовлень надходить кожного дня
- Загальну вартість робіт, яку виконав працівник для кожного замовника (перехресний запит)

Варіант 5. Реєстрація документів

- Кількість документів у кожному відділі
- Працівника, якій працює над максимальною кількістю документів
- Список документів зі зазначенням дати надходження

Варіант 6. Страхова компанія

- Список укладених договорів зі зазначенням клієнтів
- Скільки договорів уклав кожен агент
- Скільки договорів та на яку суму укладено у кожному філіалі

Варіант 7. Рецепти

- Склад рецепту – назву та які продукти необхідні для виготовлення блюда за № (№ блюда вводиться з клавіатури)
- В які блюда входить зазначений продукт - вводиться з клавіатури

- Загальну кількість кожного продукту, яка необхідна на приготування всіх страв одночасно

Варіант 8. Рейтинг пісень

- Пісні, якого композитору мають найбільший рейтинг
- Пісні якого композитора виконують зазначені виконавці (ім'я виконавця вводиться з клавіатури)
- Пісні, які займають перші три місця у рейтингу

Варіант 9. Відділ кадрів

- Які працівники прийняти на роботу за останні 10 днів
- Список працівників з вищою освітою
- Загальну кількість працівників кожної посади у кожному відділі (перехресний запит)

2.4 Створення SQL- запитів

Найбільш важливими для користувача є оператори маніпулювання даними, а саме оператор **SELECT**.

Запис операторів SQL:

- SQL –оператор складається з зарезервованих слів і слів, обумовлених користувачем;
- зарезервовані слова не можна розбивати на частині для переносу на інший рядок;
- слова, обумовлених користувачем, являють собою імена об'єктів база даних;
- слова в операторі розставляються відповідно до встановлених синтаксичних правил;
- наприкінці оператора ставиться символ;
- символні дані необхідно вводити з урахуванням регістра.

Призначення оператора **SELECT** складається у відбиранні і відображенні даних однієї чи більшої кількості таблиць бази даних. Цей оператор забезпечує виконання дій, еквівалентних операторам реляційної алгебри: вибірка, об'єднання, проекція, тощо.

Синтаксис оператора **SELECT** має наступний вигляд:

SELECT [ALL | DISTINCT] {*} [список полів [AS нове ім'я]][,...]](*)

FROM ім'я таблиці [alias][,...]

[**WHERE** Предикат-умова чи вибірки з'єднання]

[**GROUP BY** Список полів]

[**HAVING** Предикат-умова для групи]

[**ORDER BY** <Список полів, по яких упорядкувати висновок >]

Порядок пропозицій і фраз в операторі **SELECT** не може бути змінений

Ключове слово **ALL** – у результуючий набір рядків включаються всі рядки, що задовольняють умовам запиту. Без виключення дублікатів.

Ключове слово **DISTINCT** - у результуючий набір включаються тільки різні рядки, тобто дублікати рядків результату не включаються в набір.

Символ * (зірочка) - у результуючий набір включаються всі стовпці з вихідних таблиць запиту.

Приклад: Створити список студентів, із визначенням їхнього ПІБ і групи

SELECT ПІБ, Група

FROM Студенти;

Вибір рядків WHERE <Предикат-умова вибірки або з'єднання >

Для обмеження набору рядків, що поміщаються в результуючу таблицю запиту використовується пропозиція **WHERE**.

Предикати, використовувані в пропозиції **WHERE** :

- порівняння – порівнюються результати обчислень двох виразів;
- діапазон - перевіряється чи попадає результат обчислення в заданий діапазон значень -предикат **BETWEEN A AND B** — приймає значення між А и В, включаючи границі діапазону;
- приналежність до множини - перевіряється, чи належить результат до заданої множини значень -предикат входження в множину **IN** (множина), приймає значення **TRUE** коли порівнюване значення входить у множину заданих значень;
- відповідність шаблону – перевіряється, чи відповідає деяке строкове значення заданому шаблону - предикати порівняння зі зраз-

ком **LIKE i NOT LIKE** - % чи * - будь-яка послідовність символів, _ (значення одиночного символу) у шаблоні .

- Значення **NULL** - перевіряється, чи містить заданий стовець визначник **NULL**: < атрибут > **IS NULL** чи < атрибут > **IS NOT NULL**.

У пропозиції **WHERE** не можна використовувати агрегатні функції.

Приклад: визначити всіх студентів, що навчаються в групі РП-410

```
Select Прізвище
From Студенти
Where Група="РП-410";
```

У пропозиції **WHERE** можна використовувати складні умови пошуку, які реалізуються за допомогою логічних функцій **AND** або **OR**.

Приклад Перелічити усіх викладачів – доцентів кафедри Фізики
SELECT Викладачі.* - вибірка всіх полів з таблиці Викладачі
FROM Викладачі
WHERE Кафедра= «Фізики» and Посада = «Доцент»;

Приклад використання діапазонів в умовах пошуку
SELECT Викладачі.*, Викладачі.[Дата народження]
FROM Викладачі
WHERE Сесія.[Оцінка] Between 3 And 5;

Приклад використання умов пошуку з перевіркою входження в безліч (**IN / NOT IN**)

```
SELECT Викладачі.*
FROM Викладачі
WHERE Посада IN ("Доцент");
```

Приклад використання умов пошуку з указівкою шаблонів
 Список усіх співробітників проживаючих на вулиці Горького

```
SELECT Викладачі.*
FROM Викладачі
WHERE Адреса LIKE "*Горького*"; – адреса, що містить набір
```

символів Горького

Приклад використання значення NULL в умовах пошуку
SELECT Прізвище, Іспит
FROM Сесія
WHERE Оцінка is null;

Сортування результатів ORDER BY < Список полів > визначається порядок виведення результатів запиту у визначених атрибутах.

Приклад
SELECT Студенти.*,
FROM Студенти
ORDER BY Студенти.Прізвище DESC; // за зменшенням //

У SQL додані функції, що дозволяють обчислювати узагальнені групові значення. Для застосування **агрегатних функцій** передбачається попередня операція групування. При групуванні вся безліч кортежів відношення розбивається на групи, у яких збираються кортежі, що мають однакові значення атрибутів у списку групування. Агрегатні функції обчислюють одиночні значення для всієї групи таблиці

COUNT () – підраховується кількість рядків або не порожніх значень полів, що вибрав запит, застосовується для числових і нечислових полів.

SUM() – визначається сума всіх обраних значень даного поля, застосовується тільки для числових полів.

AVG() – обчислюється середнє арифметичне значення всіх обраних значень даного поля, застосовується тільки для числових полів.

MIN() – вибирається найменше з всіх обраних значень даного поля, застосовується для числових і нечислових полів.

MAX() – вибирається найбільше з всіх обраних значень даного поля, застосовується для числових і нечислових полів.

Приклад Обчислити кількість студентів, що здали успішно екзамену з конкретної дисципліни.

SELECT P1.Дисципліна, Count(*) AS Кількість
FROM P1
WHERE P1.Оцінка is not null;

Можна застосовувати агрегатні функції також і без операції по-

переднього угруповання, у цьому випадку усе відношення розглядається як одна група і для цієї групи можна обчислити одне значення на групу.

Приклад Визначити число всіх студентів, що здали успішно здали сесію

```
SELECT COUNT(*) as Кількість
FROM Сесія
WHERE Оцінка > 2;
```

Приклад Визначити скількох усього студентів здавало сесію

```
SELECT COUNT(DISTING, Код_Студента) as Кількість
FROM Сесія
WHERE Оцінка IS NOT NULL;
```

Приклад Визначити кількість усіх студентів, що здавали сесію і середній бал на сесії.

```
SELECT COUNT(DISTING, Код_Студента) as Кількість,
AVG(Оцінка) as ср_бал
FROM Сесія
WHERE Оцінка IS NOT NULL;
```

Агрегатні функції не можна використовувати в пропозиції WHERE, тому що предикати оцінюються в термінах одиночного рядка, а агрегатні функції — у термінах груп рядків.

У операторі **SELECT** можливо використання вкладених запитів, тобто зовнішній запит використовує результати внутрішнього запиту, як вихідні дані. Внутрішній запит може повертати одне значення(скалярний запит), рядок таблиці (строковий) або таблицю (табличний). Вкладений запит може використовуватись у пропозиції Select, Where, або Having.

Приклади

Скласти список всіх співробітників, які отримають заробітну платню вище за середню по підприємству та зазначити, на скільки ця зарплата більше середньої.

```
SELECT Код_робітника, ФІО, Зарплатня,
(SELECT avg(зарплатня) FROM Штат)
FROM Штат
```

WHERE Зарплатня >
(SELECT avg(зарплатня)
FROM Штат);

В цьому випадку необхідно використовувати вкладений запит, тому що не можна використовувати агрегатні функції в пропозиції **WHERE**.

Приклад

Хто з клієнтів підписав більшу кількість видань

SELECT Замовник. замовник, **count(*) as** кількість
FROM Замовник **inner join** Передплата **on** Замовник.код_замовника
 =Переплата.код_замовника
GROUP BY Замовник. замовник
HAVING Count(*)> any (Select Count(*)
from Передплата
group by Передплата.код_замовника);

2.4.1 Варіанти завдань “Створення SQL- запитів”

Інформаційна система визначена зазначеними відносинами.

- а) Визначити ключі відносин
- б) Створити SQL- запити для заданої інформаційної системи згідно варіанта

Варіант 1. Розклад

Дисципліна (Код_дисципліни, Дисципліна, Код_викладача);

Викладач (Код_викладача, Викладач);

Група (Код_групи, Група);

Розклад (Код_дисципліни, Код_групи, Код_дня, Пара, Код_виду);

Вид_занять (Код_виду, Вид_занять)

За допомогою SQL запиту визначити:

1. зайнятість викладача за тиждень (кількість занять в тиждень *2)
2. хто з викладачів працює у понеділок
3. які дисципліни читає викладач Петров І.І
4. у яких групах двічі в тиждень проводяться заняття з дисципліни

СУБД

Варіант 2. Аптека

Інформаційна система містить відносини

Група (Код_групи, Група); (антибіотики, серцеві, від кашлю, перев'язувальні матеріали і т.д.)

Ліки (Код_ліки, Ліки, Код_групи, Придатний_до)

Аптека (Код_аптеки, Аптека, Код_ліків, Кількість, Ціна, Код_постачальника)

Постачальник (Код_постачальника, Постачальник)

За допомогою SQL запиту визначити:

1. яку кількість найменувань ліків постачає кожний з постачальників
2. створити перелік ліків, для яких закінчився термін придатності
3. на яку суму в наявності кожен вид ліків у всіх аптеках міста (по кожному препарату окремо)
4. знайти аптеку, в якій вартість вказаних ліків мінімальна

Варіант 3. Передплата

Інформаційна система містить відносини

Видання (Код_видання, назв_видання, код_виду, ціна на рік)

Замовник (Код_замовника, замовник, адреса замовника)

Підписка (Код_підписки, код_видання, дата, код_періоду, код_замовника)

Період підписки (Код_періоду, період).

Вид_видання (код_виду, вид)

За допомогою SQL запиту визначити:

1. які видання, на яку суму і скільки екземплярів підписано на рік на поточну дату
2. скільки екземплярів кожного виду видання підписано
3. яке видання найдорожче
4. які клієнти підписали більше двох екземплярів на півроку
5. на який термін найчастіше підписують видання

Варіант 4. Ремонт квартир

Інформаційна система містить відносини

Замовник (Код_замовника, замовник, адреса);

Прайс_робіт (Код_роботи, вид_роботи, вартість)

Виконавець (Код_виконавця, виконавець, код_спеціальності)

Спеціальність (код_спеціальності, спеціальність)

Замовлення (Регістраційний_номер, дата, код_замовника, код_роботи, код_виконавця, термін виконання замовлення)

За допомогою SQL запиту визначити:

1. загальну суму замовлень на поточну дату
2. список працівників з вказівкою тільки вже виконаних ними робіт
3. які роботи виконуються по замовленнях частіше
4. скільки разів звернувся у фірму кожен замовник

Варіант 5. Реєстрація документів

Інформаційна система містить відносини

Виконавець (Код_виконавця, виконавець, номер_відділу)

Вид_документа (Код_документа, вид_документа)

Реєстрація (Номер_документа, Дата надходження, Код_документа, Назва документа, код_виконавця, дата_виконання)

За допомогою SQL запиту визначити:

1. які види документів зареєстровані на сьогоднішній день
2. по яких видах документів на сьогоднішній день є не оброблені документи і в якій кількості
3. список співробітників з вказівкою документів, над якими вони працюють
4. у якому відділі оброблені всі документи

Варіант 6. Страхова компанія

Інформаційна система містить відносини

Договори (Код_договору, Дата, Страхова сума, Код філіалу, Код виду страхування, код_клієнта, код_агента).

Вид страхування (Код_страхування, Страхування, Тарифна ставка).

Філіал (Код філіалу, Філіал, Адреса, Телефон)

Клієнти (Код_клієнта, Клієнт, Адреса, Телефон)

Агенти (Код_агента, Агент, Адрес_агента, Телефон,
Код_філіалу)

Філіал (Код_філіалу, Філіал, Адреса_філіалу, Телефон)

За допомогою SQL запиту визначити:

1. на яку суму поміщені договори на поточну дату
2. виплати страховим агентам за укладеними договорами (10% від страхової суми)
3. скільки поміщено договорів по кожному виду страхування
4. який договір найдорожчий

Варіант 7. Рецепти

Інформаційна система містить відносини

Блюдо (Код блюда, Назва блюда)

Постачальник (Код_постачальника, постачальник)

Продукти (Код_продукту, продукти, ціна за 1 кг, код_постачальника)

Рецепт (Код блюда, код_продукту, кількість на порцію)

За допомогою SQL запиту визначити:

1. склад рецепту блюда: назва продуктів, які входять до складу блюда, їх кількість
2. вартість кожного блюда
3. загальна кількість кожного продукту, яка необхідна для приготування всіх блюд одночасно
4. який постачальник поставляє найбільшу кількість найменувань продуктів

Варіант 8. Рейтинг пісень

Композитор (Код_композитора, композитор);

Поет (Код_поета, поет);

Виконавець (Код_виконавця, виконавець)

Рейтинг(Код_пісні, назва, код_виконавця, тривалість, код_композитора, код_поета, рейтинг пісні)

За допомогою SQL запиту визначити:

1. яка пісня має максимальний рейтинг
2. автора і виконавця найбільшої за тривалістю пісні
3. пісні, якого композитора, виконує кожен співак
4. текст, до яких пісень, написав кожен поет

Варіант 9. Відділ кадрів

Інформаційна система містить відносини

Освіта (код_ освіти, освіта)

Посада (код_ посади, посада, оклад)

Відділ (код_ відділу, відділ)

Штат(Код_ відділу, Код_ посади, кількість)

Відділ кадрів (Таб_номер, Дата прийому, ФІО, Вік, Код_ освіти, ФІО дітей, Вік дітей, будинок, телефон, код_ посади)

За допомогою SQL запиту визначити:

1. хто прийнятий на роботу за останні 10 днів
2. працівників, які отримують найменшу або найбільшу зарплату, з вказівкою її розміру
3. загальне число працівників кожного відділу
4. знайти працівників, у яких немає дітей

2.5 Контрольні питання

Відповісти на питання згідно варіанта

1. Реляційна модель організації даних
2. Ключі відношень
3. Кorteж, властивості corteжів
4. Атрибути відношень, властивості атрибутів відношень
5. Реляційна алгебра: операції над множинами – об'єднання, перетинання та вирахування
6. Реляційна алгебра: операції вилучення частки відношень - відбирання та проекція відношень
7. Нормалізація відношень: мета та послідовність виконання
8. Функціональна залежність атрибутів відношення
9. Перша нормальна форма, приведення відношення до 1НФ, навести приклад

10. Друга нормальна форма, приведення відношення до 2НФ, навести приклад
11. Третя нормальна форма, приведення відношення до 3НФ, навести приклад
12. Асоціація типу 1:1, формалізація зв'язку 1:1, навести приклад
13. Асоціація типу 1:M, формалізація зв'язку 1:M, навести приклад
14. Асоціація типу M:M, формалізація зв'язку M:M, навести приклад
15. Степінь участі сутності у зв'язку, навести приклад
16. SQL: оператор вибірки SELECT, синтаксис, використання
17. SQL: використання агрегатних функцій у запитах
18. SQL: умови відбирання рядків у запитах (ключове слово WHERE)
19. SQL: сортування та групування записів у запитах (ключові слова GROUP BY та ORDER BY)
20. SQL: умови відбирання груп записів у запитах (ключове слово HAVING)

Варіант	Номер питання		
	1	2	3
1	1	11	8
2	2	12	16
3	3	13	50
4	4	14	17
5	5	15	33
6	6	16	18
7	7	17	23
8	8	18	34
9	9	18	51
10	10	20	27

3. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

- 3.1 Конноли Т., Бегг К., Страчан А. Базы данных. Проектирование, реализация и сопровождение. Теория и практика.- Москва-Санкт-Петербург- Киев: Вильямс, 2018.-1719 с.
- 3.2 Г. Гарсия-Молина, Дж. Ульман, Дж. Уидом. Системы баз данных. Полный курс.- Москва-Санкт-Петербург - Киев: Вильямс, 2015.- 1182 с.
- 3.3 Д.Кренке. Теория и практика построения баз данных. Москва: Питер, 2016: - 850 с.