

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**

**Івано-Франківський національний технічний університет нафти і газу**

**Кафедра інженерії програмного забезпечення**

**Р. Б. Вовк**

**МЕТОДИЧНІ ВКАЗІВКИ  
ДО ВИКОНАННЯ КУРСОВОЇ РОБОТИ  
З ДИСЦИПЛІНИ “БАЗИ ДАНИХ”  
для студентів спеціальності  
121 – “Інженерія програмного забезпечення”**

**Івано-Франківськ  
2020**

УДК 004.4

ББК \_\_\_\_\_

В \_\_\_\_\_

**Рецензент:**

**Шекета В.І.** – докт. техн. наук, професор, професор кафедри інженерії програмного забезпечення ІФНТУНГ

*Рекомендовано методичною радою університету  
(протокол № \_\_\_ від «\_\_\_» 201 \_\_\_ р.)*

**Вовк Р. Б.**

В \_\_\_ Методичні вказівки до виконання курсової роботи з дисципліни “Бази даних” для студентів спеціальності 121 – “Інженерія програмного забезпечення” / Р. Б. Вовк. – Івано-Франківськ: ІФНТУНГ, 2020. – 90 с.

**МВ \_\_\_\_\_ – \_\_\_\_ – 2020**

Методичні вказівки містять постановку задачі на курсову роботу і приклад виконання одного варіанту курсової з дисципліни “Бази даних”. Постановка задачі складається з 30 варіантів комплексних завдань, до кожного з яких входять вісім задач практичного змісту. Задачі кожного з варіантів мають приблизно однакову складність. Постановка задачі в кожному окремому випадку може бути скорегована керівником курсової роботи.

Окремі задачі методичних вказівок можуть бути використаними для виконання домашніх, лабораторних і контрольних робіт студентами всіх спеціальностей. Детальне пояснення і покомандне рішення прикладу дозволяє користуватися цими методичними вказівками під час самостійного вивчення прийомів обробки баз даних засобами PostgreSQL.

Призначено для підготовки бакалаврів за спеціальністю 121 – “Інженерія програмного забезпечення” денної, заочної та дистанційної форм навчання.

**МВ \_\_\_\_\_ – \_\_\_\_ – 2020**

**УДК 004.4**

**ББК \_\_\_\_\_**

© Вовк Р. Б.

© ІФНТУНГ, 2020

	<b>ЗМІСТ</b>	стор.
	ВСТУП.....	4
1	Вимоги до тексту пояснівальної записки.....	4
1.1	Рекомендований зміст пояснівальної записки.....	4
1.2	Поради щодо складання тексту пояснівальної записки.....	4
2	Варіанти задач курсової роботи.....	5
3	Приклад виконання курсової роботи.....	41
	Рекомендована література.....	90

## ВСТУП

Курсова робота призначена для практичного оволодіння основними принципами розробки структури бази даних, усвоєння прийомів побудови таблиць, запитів, функцій та інших об'єктів, а також засвоєння навичок застосування програмних засобів сучасних систем управління базами даних.

Методичні вказівки містять постановку задачі на курсову роботу і приклад виконання. Постановка задачі складається з 30 варіантів комплексних завдань, до кожного з яких входять всім задач практичного змісту. Детальне пояснення і покомандне рішення прикладу дозволяє користуватися цими методичними вказівками під час самостійного вивчення прийомів обробки баз даних засобами PostgreSQL.

### **1. ВИМОГИ ДО ТЕКСТУ ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ**

#### **1.1. Рекомендований зміст пояснювальної записки**

**Курсова робота вважається виконаною, якщо студент представив пояснювальну записку до неї та результати роботи в електронному вигляді. Рекомендується включити в пояснювальну записку такі розділи (об'єм пояснювальної записки 35 – 50 стр.):**

Титульний лист

Зміст

Вступ

1. Постановка задачі до курсової роботи
2. Теоретична частина (короткий опис візуальних та програмних засобів PostgreSQL)
3. Виконання роботи
  - 3.1. Розробка та обґрунтування структури бази даних
  - 3.2. Побудова таблиць, підбір та обґрунтування контрольних даних таблиць
  - 3.3. Програмування запитів (застосування SQL)
4. Виготовлення звітів засобами мов PL/SQL
5. Аналіз результатів виконання програми

Висновки

Список використаної літератури

#### **1.2. Поради щодо складання тексту пояснювальної записки**

Титульний лист курсової роботи слід оформити на стандартному бланку. Друга сторінка повинна містити зміст із вказанням номерів сторінок.

Постановка задачі повинна бути ідентичною до викладеної в цих методичних вказівках.

У теоретичній частині слід коротко викласти мовні засоби, застосовані в роботі. В розділі, який містить опис програмних засобів, необхідно перелічити всі об'єкти, оператори та стандартні функції, які застосовуються в програмі, привести їхній загальний вигляд, вказати призначення, описати параметри та зробити короткий опис послідовності виконання.

Виконання роботи засобами мов PL/SQL необхідно детально відтворити, тобто потрібно записати всі команди в тій послідовності, в якій вони виконувалися. Перелік команд слід супроводжувати короткими поясненнями.

Графічне представлення алгоритмів головної програми і підпрограм (якщо вони є) необхідно виконати з дотриманням діючих стандартів. Алгоритм розв'язку задачі повинен повністю забезпечувати машинну реалізацію поставленої задачі. Він повинен бути простим, лаконічним, чітким і не містити якихось посторонніх чи неясних операцій. При поясненні

головного алгоритму необхідно відзначити звернення до підпрограм, назвати та охарактеризувати фактичні параметри.

Головна програма та підпрограми повинні повністю відповідати графічному алгоритмові. В головній програмі необхідно передбачити вивід на друк тексту, який пояснює одержані результати. Не слід засмічувати текст програми надмірною кількістю коментарів.

## 2. ВАРИАНТИ ЗАДАЧ КУРСОВОЇ РОБОТИ

### **Варіант 1 Екологічний стан атмосферного повітря за даними газохроматографічного аналізу**

Задача 1. Спроектувати та утворити базу даних Ekologia для зберігання та обробки інформації, яка містить такі дані про екологічний стан атмосферного повітря за даними газохроматографічного аналізу:

- 1– назва області України;
- 2– назва населеного пункту;
- 3– геоморфологічна прив'язка місця відбору проби;
- 4– дата відбору проби;
- 5– вміст вуглекислого газу, мг/м<sup>3</sup>;
- 6– вміст CO, мг/м<sup>3</sup>;
- 7– вміст SO<sub>2</sub>, мг/м<sup>3</sup>;
- 8– вміст бензину, мг/м<sup>3</sup>;
- 9– вміст HCl, мг/м<sup>3</sup>.

Таблиця 2.1 – Екологічний стан атмосферного повітря

1	2	3	4	5	6	7	8	9
Івано-Франківська	Підкамінь	Заплава потічка - притоки р. Свірж	22.02.97	0,033	3	0,05	1,5	0,2
Івано-Франківська	Рогатин	Заплава р. Гнила Липа	22.02.97	0,013	1,2	0,04	0,01	0
Івано-Франківська	Добринів	Берег р. Студений Потік	22.02.97	0,071	1,3	0,04	0,09	0,03
Івано-Франківська	Березівка	Вододіл	22.02.97	0,03	1,1	0	0,01	0
Чернівецька	Путила	Вододіл	22.02.97	0,031	0,2	0,01	0,01	0
Івано-Франківська	Підкамінь	Заплава потічка - притоки р. Свірж	23.02.97	0,013	1,2	0,05	0,02	0
Івано-Франківська	Рогатин	Заплава р. Гнила Липа	23.02.97	0,031	1	0	0,01	0,01
Вінницька	Вінниця	Заплава р. Південний Буг	24.03.98	0,033	0,03	0,15	1,2	0,03
Львівська	Стрий	II надзаплавна тераса р. Стрий	24.05.98	0,018	1,3	0,2	0,02	0
Чернівецька	Кіцмань	I надзаплавна тераса р. Дністер	24.07.99	0,017	1,2	0,1	0,01	0

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1– Області: код області, назва області;
- 2– Населені пункти: код населеного пункту, код області, назва населеного пункту, геоморфологічна прив'язка місця відбору проби;
- 3– Дані відбору проб: код населеного пункту, дата відбору проби, вміст вуглекислого газу, вміст CO, вміст SO<sub>2</sub>, вміст бензину, вміст HCl.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл максимальних значень вмісту SO<sub>2</sub> у атмосферному повітрі кожного населеного пункту по кварталах поточного року. Навпроти назв областей і населених пунктів установити максимальні значення вмісту речовини за кожний квартал. Передбачити вивід підсумкових максимальних значень по вертикалі для областей і по

горизонталі. До заголовка звіту додати слова “станом на” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит про середньомісячний вміст CO<sub>2</sub> у атмосферному повітрі населених пунктів выбраної області протягом поточного року.

Задача 5. Виготовити звіт у якому відобразити усереднений вміст речовин у атмосферному повітрі кожного населеного пункту протягом часу спостереження. Внести назви областей, населених пунктів та геоморфологічну прив'язку місця відбору проб. Підвести підсумки (середні значення) для кожної області (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запит з відомостями про населені пункти та про середній вміст речовин протягом часу спостереження, передбачити видачу напису “Небезпечно”, якщо середній вміст CO перевищує 1 мг/м<sup>3</sup>. Перед назвами населених пунктів додати назви областей.

Задача 7. Виготовити запити для перегляду вмісту всіх таблиць.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

## Варіант 2 Реалізація продукції хлібозаводу

Задача 1. Спроектувати та утворити базу даних Xlib для зберігання та обробки інформації, яка містить такі дані про реалізацію продукції хлібозаводу:

- 1– назва виробу;
- 2– постачальник сировини;
- 3– дата реалізації виробу;
- 4– торгова точка;
- 5– кількість, шт;
- 6– ціна за одну булку, грн;
- 7– наявність 5%-ї націнки за поставку.

Таблиця 2.2 – Реалізація продукції хлібозаводу

1	2	3	4	5	6	7
Хліб пшеничний вищого гатунку	Мілієвський мукомольний комбінат	12.06.2001	магазин № 12	200	1,25	Так
Хліб житній	Мілієвський мукомольний комбінат	03.09.2001	магазин № 19	400	1,30	Так
Хліб пшеничний вищого гатунку	Мілієвський мукомольний комбінат	04.05.2001	Кафе “Астра”	890	1,25	Hi
Хліб житній	Одесський мукомольний комбінат	23.07.2001	Бар “Веста”	450	1,20	Hi
Хліб житній з тміном	Мілієвський мукомольний комбінат	18.12.2001	магазин № 21	2000	1,30	Так
Хліб житній	Одесський мукомольний комбінат	19.04.2001	Кафе “Астра”	370	1,20	Hi
Хліб пшеничний вищого гатунку	Мілієвський мукомольний комбінат	26.01.2001	Бар “Веста”	450	1,25	Hi
Хліб пшеничний вищого гатунку	Мілієвський мукомольний комбінат	01.05.2001	магазин № 12	600	1,25	Так
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	04.03.2001	магазин № 12	1200	1,25	Так
Хліб житній з тміном	Угринівська макаронна фабрика	15.10.2001	Кафе “Астра”	1250	1,10	Hi
Булка здобна	Угринівська макаронна фабрика	18.11.2001	магазин № 19	2020	0,40	Так
Хліб житній з тміном	Мілієвський мукомольний комбінат	04.02.2000	магазин № 21	890	1,30	Так

Булка здобна	Угринівська макаронна фабрика	08.03.2000	магазин № 21	540	0,40	Так
Хліб дієтичний	Центральна товарна база	10.10.2000	магазин № 21	320	0,95	Так
Хліб житній з тміном	Мілієвський мукомольний комбінат	09.11.2000	магазин № 12	25	1,30	Так

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1– Хлібовироби: код виробу, назва виробу, ціна за одиницю виробу;
- 2– Постачальник: код постачальника, назва постачальника сировини;
- 3– Торгова точка: код торгової точки, назва торгової точки, наявність 5%-ї націнки за поставку;
- 4– Збут: код виробу, код постачальника, код торгової точки, кількість, дата.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл виручки за проданий товар у поточному році по постачальниках сировини. Навпроти назв товарів установити значення виручки для кожного постачальника. Передбачити підсумки по вертикалі і по горизонталі. Виручка дорівнює ціні, помножений на кількість товару, плюс 5%-на націнка, якщо вона є (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит, де відобразити кількість проданого житнього хліба з муки певного постачальника за кожний рік протягом останнього десятиріччя.

Задача 5. Виготовити звіт, вказати назви виробів, назви торгових точок та сумарну кількість кожного проданого товару в поточному році в тих торгових точках, де відсутня торгова націнка. До заголовка додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про збут товару.

Задача 7. Виготовити запит з відомостями про товари (назва і ціна) та постачальників сировини (назва). Відмітити товар як такий, що користується підвищеним попитом, якщо його сумарна продажа за минулий місяць перевищує 10 000 шт.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 3 Нафтогазоносні родовища України

Задача 1. Спроектувати та утворити базу даних Rodowe для зберігання та обробки інформації, яка містить такі дані про нафтогазоносні родовища України:

- 1– назва регіону;
- 2– назва області України;
- 3– назва родовища;
- 4– місце розташування родовища;
- 5– тип родовища;
- 6– розміри за покладами нафти;
- 7– розміри за покладами газу;
- 8– рік відкриття;
- 9– рік введення в розробку.

Таблиця 2.3 – Нафтогазоносні родовища України

1	2	3	4	5	6	7	8	9
Передкарпатська нафтогазоносна область	Чернівецька	Білиць-ке-Г	ст. Білиця	Нафто-газове	Середнє	Дрібне	1949	1949
Передкарпатська нафтогазоносна область	Івано-Франківська	Бого-родчанске-Г	28 км на ПдС від м. Івано-Франківськ	Газове	-	Дрібне	1963	1970

Передкарпатська нафтогазоносна область	Закарпатська	Соляти-нське-Г	1 км на Пн від м. Солятин	Нафто-газове	Дрібне	Дрібне	1982	1988
Передкарпатська нафтогазоносна область	Чернівецька	Борисів-ське-НГК	100 км на ПдЗ від м. Дробот	Нафто-газоконденсатне	Велике	Дрібне	1954	1954
Дніпровсько-Донецька нафтогазоносна область	Поставська	Аркадів-ське-ГК	35 км на С від м. Кобеляки	Газо-конденсатне	-	Середнє	1979	1980
Дніпровсько-Донецька нафтогазоносна область	Сумська	Андріївське-ГК	43 км на ПнЗ від м. Ромни	Нафто-газове	Велике	Середнє	1982	1989
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Ритнянське-Г	10 км на Пд від м. Ритня	Нафто-газове	Середнє	Дрібне	1980	1983

Виготовити таблиці БД. Рекомендації до побудови таблиць:

1 – Регіон: код регіону, назва регіону;

2 – Область: код регіону, код області, назва області України;

3 – Родовище: код області України, код родовища, назва родовища, місце розташування родовища, код типу родовища, код розміру за покладами нафти, код розміру за покладами газу, рік відкриття, рік введення в розробку;

4 – Тип родовища: код типу родовища, тип родовища;

5 – Розміри родовища: код розміру родовища, розміри родовища.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості родовищ в областях за типом родовища. Навпроти назв регіонів і областей, розділених комами, установити значення кількості для кожного типу родовища. Передбачити підсумки по вертикалі і по горизонталі. До заголовка додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит, де відобразити динаміку відкриття нових родовищ (кількість) по Україні протягом останніх восьми років.

Задача 5. Виготовити звіт з відомостями про родовища, куди внести назви регіонів, назви родовищ та різницю в роках між моментом відкриття і введення в розробку. Відібрати лише нафтогазові родовища. До заголовка додати слова “станом на ” і встановити поточну дату. Підсумувати різницю в роках для регіонів і для всього звіту (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про нові родовища.

Задача 7. Виготовити запит зі всіма відомостями про родовища, передбачити видачу відповідних повідомлень для старих родовищ, відкритих 10 і більше років тому. Вказати назви регіонів і областей.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

#### **Варіант 4 Розміри газонафтоносних родовищ України**

Задача 1. Спроектувати та утворити базу даних Nafta для зберігання та обробки інформації, яка містить такі дані про запаси нафтопродуктів у родовищах України:

1– назва регіону;

2– назва області України;

3– назва родовища;

4– місце розташування родовища;

5– розміри за покладами нафти;

- 6– розміри за вмістом газу;  
 7– рік відкриття;  
 8– рік введення в розробку.

Таблиця 2.4 – Розміри газонафтоносних родовищ України

1	2	3	4	5	6	7	8
Передкарпатська нафтогазоносна область	Чернівець-ка	Білиць-ке-Г	ст. Білиця	Середнє	Дрібне	1949	1949
Передкарпатська нафтогазоносна область	Івано-Франківська	Бого-родчанське-Г	28 км на ПдС від м. Івано-Франківськ	-	Дрібне	1963	1970
Передкарпатська нафтогазоносна область	Закарпатська	Солятичнське-Г	1 км на Пн від м. Солятин	Дрібне	Дрібне	1982	1988
Передкарпатська нафтогазоносна область	Чернівець-ка	Борисівське-НГК	100 км на ПдЗ від м. Дробот	Велике	Дрібне	1954	1954
Дніпровсько-Донецька нафтогазоносна область	Поставська	Аркадівське-ГК	35 км на С від м. Кобеляки	-	Середнє	1979	1980
Дніпровсько-Донецька нафтогазоносна область	Сумська	Андріївське-ГК	43 км на ПнЗ від м. Ромни	Велике	Середнє	1982	1989
Прічорноморсько-Кримська газонафтоносна впадина	Одеська	Ритнянське-Г	10 км на Пд від м. Ритня	Середнє	Дрібне	1980	1983

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області України;
- 3 – Родовище: код родовища, код області України, назва родовища, місце розташування родовища, код розміру за покладами нафти, код розміру за покладами газу, рік відкриття, рік введення в розробку;
- 4 – Розміри родовища: код розміру родовища, розміри родовища.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості родовищ за розмірами за покладами нафти по областях. Навпроти назв областей установити значення кількості родовищ для кожного розміру за покладами нафти. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит, відобразити динаміку (кількість) введення родовищ у розробку по Україні протягом останніх 10 років.

Задача 5. Виготовити звіт, куди внести назви регіонів і родовищ, відкритих протягом останніх 10 років, рік відкриття та розміри за покладами нафти. Відібрати лише великі та середні родовища за покладами нафти. До заголовка додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про нові родовища.

Задача 7. Виготовити запит зі всіма відомостями про родовища, передбачити видачу відповідних повідомлень для родовищ, введених у розробку за останні 5 років. Вказати назви регіонів і областей.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

#### Варіант 5 Газосховища України

Задача 1. Спроектувати та утворити базу даних Gas для зберігання та обробки інформації, яка містить такі дані про газосховища України:

- 1– назва комплексу;
- 2– назва газового сховища;
- 3– об’єм газового сховища, млн м<sup>3</sup>;
- 4– кількість свердловин;
- 5– добова продуктивність свердловин;
- 6– кількість запомпованого газу, млн. м<sup>3</sup>;
- 7– кількість вибраного газу, млн. м<sup>3</sup>;
- 8– дата занесення даних.

Таблиця 2.5 – Газосховища України

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Західно-Український	Луцьке	8150	12	12	1937,6	210,4	04.04.1990
Західно-Український	Черепівське	1000	33	19	118,1	396,1	05.06.1990
Західно-Український	Черепівське	1000	33	19	198,2	120,2	07.09.1989
Київський	Тетіївське	4000	147	8	631,3	81,2	23.09.1990
Київський	Тетіївське	4000	147	8	678,7	72,4	12.08.1991
Київський	Таращанське	1800	34	3	915,9	343,2	20.09.1988
Київський	Таращанське	1800	34	3	880,0	34,1	12.11.1990
Київський	Таращанське	1800	34	3	310,5	310,0	23.10.1988
Донецький	Володарське	300	57	5	120,2	42,4	26.09.1991
Донецький	Крапське	2000	66	5	352,5	368,5	17.11.1989
Південно-Український	Каховське	1740	81	6	573,2	387,4	15.12.1991
Південно-Український	Новоліське	1235	55	7	456,0	345,0	12.09.1992
Південно-Український	Новоліське	1235	55	7	654,0	450,0	15.06.1993

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1– Комплекс: код комплексу, назва комплексу;
- 2– Сховище: код комплексу, код газового сховища, назва газового сховища, об’єм газового сховища, кількість свердловин, добова продуктивність свердловин;
- 3– Рух газу: код газового сховища, кількість запомпованого газу, кількість вибраного газу, дата занесення даних.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості вибраного газу подекадно для поточного місяця. Навпроти назв комплексів і сховищ, розділених пробілом, установити значення кількості за кожну декаду. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит, де показати динаміку річної зміни вмісту газу (вміст дорівнює кількості запомпованого газу мінус кількість вибраного газу) в сховищах протягом останніх десяти років.

Задача 5. Виготовити звіт для заданого комплексу про кількість запомпованого газу у кожне сховище на поточну дату. До заголовка звіту додати слова “станом на” і покласти поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць та доповнення новими відомостями таблиці з даними про рух газу.

Задача 7. Виготовити запит з відомостями про сховища, передбачити видачу відповідних повідомлень для малих сховищ, де кількість свердловин менша за 100. Вказати назви комплексів.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 6 Нафтобази України

Задача 1. Спроектувати та утворити базу даних Naftobaza для зберігання та обробки інформації, яка містить такі дані про нафтобази України:

- 1– назва регіону;
- 2– назва області;
- 3– назва нафтобази;
- 4– кількість резервуарів;
- 5– сумарний об'єм резервуарів, м<sup>3</sup>;
- 6– кількість нафтопродукту, тис. тон;
- 7– дата обліку;

Таблиця 2.6 – Нафтобази України

1	2	3	4	5	6	7
Передкарпатська нафтогазоносна область	Івано-Франківська	Снятин НБ	28	93053	188,200	12.08.1997
Передкарпатська нафтогазоносна область	Закарпатська	Майдан НБ	36	9479	80,060	09.06.1995
Передкарпатська нафтогазоносна область	Закарпатська	Чоп НБ	37	16936	77,432	23.08.1991
Передкарпатська нафтогазоносна область	Чернівецька	Вижниця НБ	210	18114	30,400	20.12.1995
Передкарпатська нафтогазоносна область	Чернівецька	Чернівці НБ	16	16902	138,150	23.11.1998
Передкарпатська нафтогазоносна область	Львівська	Дубляни НБ	35	12693	87,641	15.12.1990
Передкарпатська нафтогазоносна область	Львівська	Ходорів НБ	110	347190	0,002	18.06.1990
Передкарпатська нафтогазоносна область	Львівська	Яворів НБ	320	172652	63,380	28.07.1990
Дніпровсько-Донецька нафтогазоносна область	Луганська	Мілове НБ	44	3066	28,700	17.03.1993
Дніпровсько-Донецька нафтогазоносна область	Харківська	Есхар НБ	156	15675	24,711	25.11.1997
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Сарата НБ	40	15610	60,400	15.12.1990
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Тузли НБ	308	108405	44,400	08.03.1992
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Городнє НБ	25	7526	28,800	06.07.1998
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Берізка НБ	333	12755	24,400	02.04.1998
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Далеке НБ	18	8928	41,300	12.02.1999
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Судак НБ	33	10527	52,440	19.10.1991
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Бахчи НБ	25	7053	49,900	11.12.1991

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області ;
- 3 – Нафтобаза: код області, код нафтобази, назва нафтобази, кількість резервуарів, об’єм резервуарів;
- 4 – Нафтопродукти: код нафтобази, кількість нафтопродукту, дата обліку.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про попіврічний розподіл кількості нафтопродукту в певному регіоні в минулому році. Навпроти назв областей і нафтобаз, розділених пробілом, установити значення кількості за кожне півріччя. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит, показати динаміку річної ефективності використання резервуарів по Україні протягом останніх 5 років, яка дорівнює кількості нафтопродукту, поділеній на об’єм резервуарів.

Задача 5. Виготовити звіт про поточний вміст нафтопродуктів у заданому регіоні. До його заголовка додати слова “станом на” і встановити поточну дату. Врахувати, що дані в базу поступають нерегулярно, тому в звіт необхідно внести останні дані (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про рух нафтопродуктів.

Задача 7. Виготовити запит з відомостями про нафтобази, передбачити видачу відповідних повідомлень для нафтобаз з кількістю резервуарів більшою за 300. Вказати назви регіонів і областей.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 7 Транспортування газу по території України**

Задача 1. Спроектувати та утворити базу даних Transgas для зберігання та обробки інформації, яка містить такі дані про транспортування газу по території України:

- 1– назва газопроводу;
- 2– кількість ниток газопроводу;
- 3– діаметр однієї нитки, мм;
- 4– назва підприємства, яке експлуатує газопровід;
- 5– добова кількість запомпаного газу, м<sup>3</sup>;
- 6– дата представлення даних.

Таблиця 2.7 – Транспортування газу по території України

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>
Острогожськ - Гадяч	2	1000	Гадячгазпром	15537,6	08.07.1990
Київ – Кіцмань	1	1000	Київтрансгаз	15220,1	04.03.1990
Новопськовськ - Гадяч	1	1200	Гадячгазпром	9726,6	17.05.1991
Острогожськ - Гадяч	2	1000	Гадячгазпром	15537,6	19.04.1995
Ямпіль - Ужгород - держкордон	2	1400	Прикарпаттрансгаз	26787,5	25.12.1994
Ямпіль - Ужгород - держкордон	2	1400	Прикарпаттрансгаз	26787,5	23.04.1993
Гадяч - Кобеляки - Київ	1	700	Київтрансгаз	3354,8	08.09.1997
Єлецьк – Орел	1	1200	Харківтрансгаз	24867,5	04.07.1995
Київ – Кіцмань	1	1000	Львівтрансгаз	15220,1	12.06.1993
Київ – Кіцмань	1	1000	Прикарпаттрансгаз	15220,1	28.10.1992
Новопськовськ - Гадяч	1	1200	Експорттрансгаз	9726,6	22.11.1994

Гадяч – Ізмаїл	3	1000	Харківтрансгаз	29484,9	09.12.1991
Москва - Каховка - Ужгород	3	1400	Київтрансгаз	49926,9	01.04.1991
Єлецьк - Десна - Кривий Ріг	2	1400	Київтрансгаз	28489,7	08.04.1992
Орел – Київ	1	1200	Харківтрансгаз	8458,8	03.02.1993
Москва - Каховка - Ужгород	3	1400	Прикарпаттрансгаз	49926,0	12.12.1992
Москва - Каховка - Ужгород	3	1400	Експорттрансгаз	49926,7	17.11.1992
Москва - Каховка - Ужгород	3	1400	Харківтрансгаз	49926,0	19.10.1991
Україна	1	1400	Прикарпаттрансгаз	22914,9	05.10.1994
Єлецьк - Десна - Кривий Ріг	2	1400	Експорттрансгаз	28489,7	10.06.1993
Єлецьк - Десна - Кривий Ріг	2	1400	Харківтрансгаз	28489,7	27.04.1996
Україна	1	1400	Експорттрансгаз	22914,9	15.09.1994
Москва - Каховка - Ужгород	3	1400	Прикарпаттрансгаз	51955,4	18.06.1991
Івшевичі - Ямпіль	2	1200	Львівтрансгаз	18893,1	12.09.1992
Орел – Київ	1	1200	Львівтрансгаз	8545,4	16.04.1993
Москва - Каховка - Ужгород	3	1400	Експорттрансгаз	51955,4	18.02.1994
Москва - Каховка - Ужгород	3	1400	Харківтрансгаз	51955,4	12.12.1995
Ямпіль - Ужгород - держкордон	2	1400	Прикарпаттрансгаз	19784,6	19.05.1996
Івшевичі - Париж	1	800	Львівтрансгаз	5253,4	17.05.1994
Гадяч - Кобеляки - Київ	1	700	Київтрансгаз	4127,2	03.01.1998
Україна	1	1400	Прикарпаттрансгаз	26550,9	06.04.1992
Орел – Київ	1	1200	Харківтрансгаз	7571,3	28.01.1996

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Підприємство: код підприємства, назва підприємства, яке експлуатує газопроводи;
- 2– Газопровід: код газопроводу, назва газопроводу, кількість ниток газопроводу, діаметр однієї нитки.
- 3– Облік газу: код газопроводу, код підприємства, поточна кількість перепомпованого газу за добу, м<sup>3</sup>, дата представлення даних;

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості транспортованого газу на кожному підприємстві по газопроводах у поточному році. Навпроти назв підприємств установити значення кількості для кожного газопроводу. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про сумарну кількість транспортованого газу за кожний рік протягом останніх 5 років для заданого газопроводу.

Задача 5. Виготовити звіт про сумарний об'єм транспортованого газу за поточний рік по кожному газопроводу для заданого підприємства. До заголовка звіту додати назву газопроводу, слова “станом на” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про рух газу. Передбачити перегляд таких даних: назва підприємства, назва газопроводу, кількість ниток газопроводу, діаметр газопроводу, дата та кількість перепомпованого газу.

Задача 7. Виготовити запит з відомостями про газопроводи (назва, кількість ниток, діаметр однієї нитки, мм), передбачити видачу відповідних повідомлень для газопроводів, яких діаметр однієї нитки дорівнює 1,4 м та більше. Вказати сумарну кількість транспортованого газу в м куб. протягом періоду експлуатації.

Задача 8. Скласти програму для формування і видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 8 Забезпеченість України паливно-мастильними матеріалами**

Задача 1. Спроектувати та утворити базу даних (БД) Palyvo для зберігання та обробки інформації, яка містить такі дані про забезпеченість паливно-мастильними матеріалами:

- 1– назва регіону;
- 2– назва області;
- 3– назва нафтобази;
- 4– дата представлення даних;
- 5– кількість бензину, тис. т;
- 6– кількість дизпалива, тис. т;
- 7– кількість машинного масла, тис. т.

Таблиця 2.8 – Забезпеченість паливно-мастильними матеріалами

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
Передкарпатська нафтогазоносна область	Львівська	Броди НБ	09.01.1993	28,800	28,600	1,400
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Гадяч НБ	19.04.1994	57,188	74,348	0,826
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Глобине НБ	23.01.1992	11,540	12,041	0,253
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Піски НБ	24.02.1996	8,918	11,698	0,144
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Кобеляки НБ	09.04.1991	81,000	62,500	9,700
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Котельва НБ	05.05.1997	29,900	32,900	1,400
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Чутове НБ	03.07.1998	32,631	38,440	3,136
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Миргород НБ	09.03.1992	6,490	13,955	1,332
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Алушта НБ	06.04.1999	10,700	18,900	0,900
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Ялта НБ	18.02.1990	51,200	31,200	1,100
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Судак НБ	22.05.1993	43,600	58,500	1,200
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Котельва НБ	06.04.1999	9,070	3,980	8,900
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Ялта НБ	03.07.1998	12,678	89,000	0.800
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Миргород НБ	18.02.1990	89,982	45,050	0,765
Передкарпатська нафтогазоносна область	Івано-Франківська	Одаї НБ	09.03.1992	22,087	13,089	0,500

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області;
- 3 – Нафтобаза: код області, код нафтобази, назва нафтобази;

4 – Паливо: код нафтобази, дата представлення даних, кількість бензину, кількість дизпалива, кількість машинного масла.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про поквартальний розподіл вмісту бензину у нафтобазах. Навпроти назв регіонів, областей і нафтобаз, розділених пробілом, установити значення вмісту бензину за кожний квартал поточного року. Передбачити підсумки по горизонталі та по вертикалі для регіонів і всього звіту (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про сумарний вміст дизпалива протягом останніх 10 років у нафтобазах заданої області.

Задача 5. Виготовити звіт про сумарний вміст нафтопродуктів у кожній області. Підвести підсумки по регіонах. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про рух нафтопродуктів.

Задача 7. Виготовити запит з відомостями про нафтобази, передбачити видачу повідомлення “Карпати” для нафтобаз, розташованих в Івано-Франківській або Чернівецькій областях. Вказати назви регіонів та областей.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 9 Видобування нафти на Україні**

Задача 1. Спроектувати та утворити базу даних Dobuv для зберігання та обробки інформації, яка містить такі дані про видобування нафти на Україні:

- 1– назва нафтогазоносного регіону;
- 2– назва області;
- 3– назва наftового родовища;
- 4– дата представлення даних;
- 5– адреса родовища;
- 6– рік введення родовища в експлуатацію;
- 7– видобуток нафти, тис. т;
- 8– кількість свердловин родовища.

Таблиця 2.9 – Видобування нафти на Україні

1	2	3	4	5	6	7	8
Дніпровсько-Донецький	Херсонська	Каховське-Н	12.09.1995	5 км на ПдС від м. Херсон	1990	6120	12
Дніпровсько-Донецький	Херсонська	Каховське-Н	04.05.1997	5 км на ПдС від м. Херсон	1990	970	12
Дніпровсько-Донецький	Полтавська	Хорольсь-ке-НГК	03.08.1995	9 км на ПдС від м. Кобеляки	1964	58766	25
Дніпровсько-Донецький	Дніпропетровська	Могилівське-НГ	17.12.1995	3 км від м. Новомосковськ	1976	4590	43
Дніпровсько-Донецький	Полтавська	Хорольсь-ке-НГК	16.10.1995	9 км на ПдС від м. Кобеляки	1964	2250	25
Дніпровсько-Донецький	Дніпропетровська	Роздорівське-НГК	07.12.1995	21 км на ПнЗ від с. Іскра	1985	3050	23
Дніпровсько-Донецький	Сумська	Левадівське-НГК	07.03.1999	100 км на ПнС від м. Ромни	1973	6787	35
Дніпровсько-Донецький	Сумська	Корецьке-НГК	28.05.1995	50 км на ПнЗ від м. Ромни	1990	31575	64
Дніпровсько-Донецький	Сумська	Корецьке-НГК	22.11.1996	50 км на ПнЗ від м. Ромни	1990	804161	64

Передкарпатський	Івано-Франківська	Коломийське-Н	11.12.2000	60 км на П від м. Ів-Франківськ	1956	28459	34
Передкарпатський	Івано-Франківська	Калуське-НГК	23.12.1997	30 км на З від м. Ів-Франківськ	1962	7108	37
Передкарпатський	Львівська	Ходорівське-НГК	29.01.1993	10 км на ПдЗ від м. Львів	1954	59025	45
Передкарпатський	Львівська	Глиннянське-Н	28.02.1998	20 км на ПдЗ від м. Бібрка	1967	1934	22
Передкарпатський	Львівська	Стрийське-Н	22.09.1995	15 км на ПдЗ від м. Стрий	1890	4527	18
Передкарпатський	Івано-Франківська	Тлумацьке-Н	17.09.1995	7 км на ПдС від м. Ів-Франківськ	1967	9164	35
Передкарпатський	Івано-Франківська	Тлумацьке-Н	19.07.1997	7 км на ПдС від м. Ів-Франківськ	1967	17384	35

Виготовити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код нафтогазоносного регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області;
- 3 – Родовище: код області, код родовища, назва нафтового родовища, адреса родовища, рік введення родовища в експлуатацію, кількість свердловин родовища;
- 4 – Видобуток: код родовища, дата представлення даних, видобуток нафти.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості родовищ в областях по числу свердловин. Родовища повинні бути розділені на три групи, які матимуть до 30, від 31 до 60 та від 61 до 90 свердловин. Навпроти назв регіонів і областей, розділених пробілом, установити значення кількості родовищ для кожної групи. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про сумарний видобуток нафти у заданій області за кожний місяць поточного року.

Задача 5. Виготовити звіт про середньомісячний видобуток нафти у кожному регіоні протягом останніх десяти років. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про видобуток нафти.

Задача 7. Виготовити запит з відомостями про родовища, передбачити видачу відповідних повідомлень для тих родовищ, де видобуток нафти перевищує 100 тис. т. Вказати назви регіонів та областей.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 10 Запаси нафти на родовищах України

Задача 1. Спроектувати та утворити базу даних Zapas для зберігання та обробки інформації, яка містить такі дані про запаси нафти на родовищах України:

- 1– назва регіону;
- 2– назва області;
- 3– назва родовища;
- 4– категорія родовища;
- 5– геологічні запаси, тис. т;
- 6– видобувні запаси, тис. т;
- 7– дата представлення даних.

Таблиця 2.10 – Запаси нафти на родовищах України

1	2	3	4	5	6	7
---	---	---	---	---	---	---

Передкарпатська нафтогазоносна область	Івано-Франківська	Коломийське-Н	В	1593	129	19.04.94
Передкарпатська нафтогазоносна область	Івано-Франківська	Слобідське-Н	А	775	3871	21.09.93
Передкарпатська нафтогазоносна область	Івано-Франківська	Снятинське-ГН	AB	726	13245	11.09.93
Передкарпатська нафтогазоносна область	Чернівецька	Лужанське-Н	C2	6169	1814	18.03.92
Передкарпатська нафтогазоносна область	Чернівецька	Хотинське-Н	ABC1	16844	4897	19.06.02
Передкарпатська нафтогазоносна область	Львівська	Жидачівське-Н	C1	962	9935	10.03.03
Дніпровсько-Донецька нафтогазоносна область	Херсонська	Бехтерське-Н	ABC2	117	462	01.09.93
Дніпровсько-Донецька нафтогазоносна область	Херсонська	Пнівське-НГК	AB	903	313	21.07.02
Дніпровсько-Донецька нафтогазоносна область	Херсонська	Пнівське-НГК	AB	903	36	09.06.04
Дніпровсько-Донецька нафтогазоносна область	Сумська	Східнівське-НГ	В	1483	23	07.10.04
Дніпровсько-Донецька нафтогазоносна область	Сумська	Східнівське-НГ	В	1483	184	04.04.04
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Жовтярське-Н	C1	2292	500	03.03.02
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Жовтярське-Н	C1	2292	340	02.02.03
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Сівське-Н	A	254	14	11.19.94
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Сівське-Н	A	254	241	15.11.02
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Сівське-Н	A	254	14	21.12.92

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області;
- 3 – Родовище: код області, код родовища, назва родовища, геологічні запаси, категорія родовища;
- 4 – Запаси: код родовища, видобувні запаси, дата представлення даних.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості родовищ областей по категоріях. Навпроти назв регіонів і областей, розділених пробілом, установити значення кількості родовищ для кожної категорії. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку зміни видобувних запасів нафти протягом останніх 15 років заданого родовища. Додати назву цього родовища.

Задача 5. Виготовити звіт про різницю геологічних і видобувних запасів нафти в родовищах на поточну дату. Оскільки зміна видобувних запасів відбувається нерегулярно (рідко), врахувати останні дані. У звіт внести скорочені назви регіонів та повні назви областей і родовищ, розділені комами. До заголовка звіту додати слова “станом на” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць зі всіма даними та доповнення БД відомостями про зміну видобувних запасів нафти.

Задача 7. Виготовити запит з відомостями про родовища: назва регіону, назва області, назва родовища, категорія родовища, геологічні та видобувні запаси нафти. Вказати, що родовище невелике, якщо геологічні запаси нафти менші за 1 млн т.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 11 Зміни запасів нафти в родовищах України

Задача 1. Спроектувати та утворити базу даних Zmina для зберігання та обробки інформації, яка містить такі дані про зміни запасів нафти в родовищах України:

- 1– назва регіону;
- 2– назва області;
- 3– назва нафтогазоносного родовища;
- 4– дата представлення даних;
- 5– початкові запаси нафти, тис. т;
- 6– середньодобовий видобуток нафти на представлена дату, тис. т.

Таблиця 2.11 – Зміни запасів нафти в родовищах України

1	2	3	4	5	6
Дніпровсько-Донецька нафтогазоносна область	Сумська	Канівське-НГК	12.07.1999	1619,1	92
Дніпровсько-Донецька нафтогазоносна область	Сумська	Рабарське-НГК	23.05.1994	614,7	38
Передкарпатська нафтогазоносна область	Львівська	Борисівське-НГК	14.04.1998	212,0	50
Передкарпатська нафтогазоносна область	Львівська	Орловське-Н	29.03.1993	315,5	14
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Семенівське-Н	12.10.1993	1115,9	3
Дніпровсько-Донецька нафтогазоносна область	Харківська	Козівське-Н	11.11.1994	112,6	25
Дніпровсько-Донецька нафтогазоносна область	Харківська	Захівське-Н	09.11.1994	11,8	3
Передкарпатська нафтогазоносна область	Івано-Франківська	Черпинське-Н	04.10.1997	101,1	1
Передкарпатська нафтогазоносна область	Івано-Франківська	Підлісівське-Н	15.01.1996	1,9	1
Передкарпатська нафтогазоносна область	Чернівецька	Лобунянське-Н	28.02.1994	300,2	51
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Семенівське-Н	17.05.1994	1115,9	1
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Розлоге-Н	23.07.1995	565,0	123
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Розлоге-Н	12.04.1998	565,0	34

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Регіон: код регіону, назва регіону;
- 2 – Область: код регіону, код області, назва області;
- 3 – Родовище: код області, код родовища, назва нафтогазоносного родовища, початкові запаси;
- 4 – Запаси: код родовища, дата представлення даних, середньодобовий видобуток нафти.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості родовищ в областях по початкових запасах нафти. Родовища розділити на 3 групи з початковими запасами нафти до 1000, 1001-2000 і більше 2000 тис. т. Навпроти назв областей і регіонів, розділених пробілом, установити значення кількості для кожної групи. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку зменшення запасів нафти за кожний місяць минулого року в заданій області. Вказати назву області.

Задача 5. Виготовити звіт про залишкові запаси нафти на кожному родовищі на поточну дату. Залишкові запаси нафти дорівнюють початковим запасам мінус сумарний середньодобовий видобуток нафти протягом періоду експлуатації. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про видобуток нафти.

Задача 7. Виготовити запит зі всіма відомостями про родовища, включаючи назви регіонів і областей. Позначити відповідним написом великі родовища, початкові запаси яких перевищують 1000 тис. т.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 12 Гідрохімічний стан поверхневих вод**

Задача 1. Спроектувати та утворити базу даних Stanvod для зберігання та обробки інформації, яка містить такі дані про гідрохімічний стан поверхневих вод:

- 1– назва населеного пункту;
- 2– геоморфологічна прив’язка;
- 3– дата взяття проби;
- 4– кількість завислих речовин, мг/дм<sup>3</sup>;
- 5– значення pH;
- 6– кількість NH<sub>4</sub>, мг/дм<sup>3</sup>;
- 7– вміст нітратів, мг/дм<sup>3</sup>;
- 8– вміст хлоридів, мг/дм<sup>3</sup>.

Таблиця 2.12 – Гідрохімічний стан поверхневих вод

1	2	3	4	5	6	7	8
Підкамінь	Заплава потічка - притоки р. Свірж	01.02.1990	655	7	0,64	0,1	0,9
Підкамінь	Заплава потічка - притоки р. Свірж	04.05.1991	147	7	1,16	0,2	0,1
Підкамінь	Заплава потічка - притоки р. Свірж	04.05.1992	86	7	1,16	0,2	0,1
Наставино	Схил долини р. Гнила Липа	04.05.1992	44	7	1,84	0,0	0,9
Куропатники	Заплава р. Гнила Липа	01.02.1992	120	7	1,54	0,0	0,7
Куропатники	Заплава р. Гнила Липа	04.05.1994	154	6,5	2,65	0,4	3,6
Куропатники	Заплава р. Гнила Липа	15.09.1995	556	6,4	3,94	0,6	4,9
Наставино	Схил долини р. Гнила Липа	04.05.1996	336	7	1,16	0,1	1,3
Наставино	Схил долини р. Гнила Липа	05.08.1997	256	7	1,15	0,1	1,3
Журавський	ІІІ надзаплавна тераса р. Дністер	04.05.1998	134	7	0,64	0,0	1,4
Журавський	ІІІ надзаплавна тераса р. Дністер	22.10.1997	147	7	0,29	0,0	1,7
Чагрів	Вододіл	04.05.1997	224	7	0,65	0,0	1,2
Наставино	Схил долини р. Гнила Липа	03.06.1997	664	6,5	3,11	0,1	4,2
Поділля	Заплава потічка	04.05.1997	361	6,4	2,89	0,3	3,9
Підшумлянці	Схил долини р. Нараївки	04.05.1997	121	7	0,11	0,0	1,7
Довжка	Вододіл	04.05.1997	247	7	0,33	0,0	1,3

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Населений пункт: код населеного пункту, назва населеного пункту, геоморфологічна прив’язка;
- 2– Вміст елементів: код населеного пункту, дата взяття проби, кількість завислих речовин, значення pH, кількість NH<sub>4</sub>, вміст нітратів, вміст хлоридів.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про подекадний розподіл середнього вмісту завислих речовин у населених пунктах за поточний місяць поточного року. Навпроти назв населених пунктів і

геоморфологічної прив'язки, розділених пробілом, установити середні значення вмісту речовин за кожну декаду. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит про динаміку зміни максимального вмісту хлоридів у кварталах минулого року для заданого населеного пункту.

Задача 5. Виготовити звіт про середньорічний вміст речовин у кожному населеному пункті протягом часу спостереження. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про вміст елементів.

Задача 7. Виготовити запит з відомостями про населені пункти, вказати назву населеного пункту, геоморфологічну прив'язку та середній вміст елементів протягом поточного року. Забезпечити відповідний напис, якщо середня кількість завислих речовин перевищує 200 мг/дм<sup>3</sup>.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 13 Реалізація продукції хлібозаводу

Задача 1. Спроектувати та утворити базу даних Zavod для зберігання та обробки інформації, яка містить такі дані про реалізацію продукції хлібозаводу:

- 1– назва виробу;
- 2– постачальник сировини;
- 3– дата реалізації виробу;
- 4– торгова точка;
- 5– кількість, шт;
- 6– ціна за одну булку, грн;
- 7– наявність 5%-ї націнки за поставку.

Таблиця 2.13 – Реалізація продукції хлібозаводу

1	2	3	4	5	6	7
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	12.06.2001	магазин № 12	200	1,25	Так
Хліб житній	Одеський мукомольний комбінат	03.09.2001	магазин № 19	400	1,20	Так
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	04.05.2001	Кафе “Астра”	890	1,25	Так
Хліб житній	Одеський мукомольний комбінат	23.07.2001	Бар “Веста”	450	1,20	Ні
Хліб житній з тміном	Мілієвський мукомольний комбінат	18.12.2001	магазин № 21	2000	1,30	Ні
Хліб житній	Одеський мукомольний комбінат	19.04.2001	Кафе “Астра”	370	1,20	Ні
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	26.01.2001	Бар “Веста”	450	1,25	Так
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	01.05.2001	магазин № 12	600	1,25	Так
Хліб пшеничний вищого сорту	Мілієвський мукомольний комбінат	04.03.2001	магазин № 12	1200	1,25	Так
Хліб житній з тміном	Мілієвський мукомольний комбінат	15.10.2001	Кафе “Астра”	1250	1,30	Ні
Булка здобна	Угринівська макаронна фабрика	18.11.2001	магазин № 19	2020	0,40	Ні
Хліб житній з тміном	Мілієвський мукомольний комбінат	04.02.2000	магазин № 21	890	1,30	Так
Булка здобна	Угринівська макаронна фабрика	08.03.2000	магазин № 21	540	0,40	Ні
Хліб дієтичний	Центральна товарна база	10.10.2000	магазин № 21	320	0,95	Так
Хліб житній з тміном	Мілієвський мукомольний комбінат	09.11.2000	магазин № 12	25	1,30	Ні

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Хлібовироби: код виробу, назва виробу, код постачальника сировини, ціна за одиницю;

- 2– Постачальник: код постачальника, назва постачальника сировини;  
 3– Торгова точка: код торгової точки, назва торгової точки;  
 4– Збут: код виробу, код торгової точки, кількість, дата, наявність 5%-ї націнки за поставку.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл виручки за проданий товар по торгових точках у поточному році. Навпроти назв товарів і цін, розділених пробілом, установити значення виручки в кожній торговій точці. Передбачити підсумки по вертикалі і по горизонталі. Виручка дорівнює ціні, помноженій на кількість товару, плюс 5%-на націнка, якщо вона є (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит, де відобразити продажу заданого продукту за кожний рік протягом останнього десятиріччя.

Задача 5. Виготовити звіт про продані товари, вказати назви виробів, назви торгових точок та сумарну кількість проданого товару в тих випадках, де була торгова націнка. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про реалізацію товару.

Задача 7. Виготовити запит з відомостями про товари та постачальників сировини. Відмітити товар як такий, що користується підвищеним попитом, якщо його середньомісячна продажа за останній рік перевищує 10 000 шт.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

#### **Варіант 14 Облік відвідування занять студентами вузу**

Задача 1. Спроектувати та утворити базу даних Vidvid для зберігання та обробки інформації, яка містить такі дані про облік відвідування занять студентами університету:

- 1– група;
- 2– прізвище;
- 3– ім’я;
- 4– дата обліку;
- 5– кількість пропущених годин;
- 6– причина (поважна/неповажна);
- 7– адреса студента.

Таблиця 2.14 – Облік відвідування занять студентами

1	2	3	4	5	5	7
АА-00-01	Петренко	Петро	12.06.00			гуртожиток № 3
АА-00-01	Семенченко	Семен	12.06.00			гуртожиток № 3
АА-00-01	Федоренко	Федір	12.06.00	6	поважна	гуртожиток № 3
АА-00-01	Стеценко	Василь	12.06.00			вул. Зелена, 6
АА-00-01	Мазур	Михайло	12.06.00			гуртожиток № 3
AB-99-02	Семенченко	Степан	12.06.00	4	неповажна	гуртожиток № 4
AB-99-02	Деркач	Іван	12.06.00			смт. Дуляни
BC-99-02	Притула	Дмитро	12.06.00			вул. Петренка, 12
BC-99-02	Барка	Ніна	12.06.00			вул. Польова, 3
BC-99-02	Завійський	Анатолій	12.06.00	6	поважна	гуртожиток № 4
AC-99-02	Сливка	Ольга	12.06.00			пер. Крутий, 7
AA-00-01	Петренко	Петро	13.06.00			гуртожиток № 3
AA-00-01	Семенченко	Семен	13.06.00	3		гуртожиток № 3
AA-00-01	Федоренко	Федір	13.06.00			гуртожиток № 3
AA-00-01	Стеценко	Василь	13.06.00	6	поважна	вул. Зелена, 6
AA-00-01	Мазур	Михайло	13.06.00			гуртожиток № 3
AB-99-02	Семенченко	Степан	13.06.00			гуртожиток № 4

AB-99-02	Деркач	Іван	13.06.00			смт. Дуляни
BC-99-02	Притула	Дмитро	13.06.00	6	неповажна	вул. Петренка, 12
BC-99-02	Барка	Ніна	13.06.00			вул. Польова, 3
BC-99-02	Завійський	Анатолій	13.06.00			гуртожиток № 4
AC-99-02	Сливка	Ольга	13.06.00	2	неповажна	пер. Крутий, 7
BC-99-02	Завійський	Анатолій	14.06.00			гуртожиток № 4
AC-99-02	Сливка	Ольга	14.06.00	1		пер. Крутий, 7

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Студенти: код групи, код студента, прізвище, ім’я, адреса студента;
- 2– Пропуски: код студента, кількість пропущених годин, причина, дата обліку;
- 3– Група: назва групи, код групи.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл сумарної кількості пропусків з неповажних причин у групах по місяцях поточного семестру. Врахувати, що перший семестр починається у вересні, а другий – у лютому, тривалість семестру дорівнює 4 місяці. У звіті відобразити дані від початку семестру до поточного місяця включно. Навпроти назв груп установити значення кількості пропусків за кожний місяць. Передбачити підсумки по вертикалі і по горизонталі. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про зміни сумарної кількості пропусків у заданій групі за кожний навчальний рік. Термін навчання студента дорівнює 5 років.

Задача 5. Виготовити звіт про студентів, які мали більше як 250 пропущених годин з неповажних причин за весь період навчання. До звіту внести прізвище, ім’я та адресу (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про пропуски.

Задача 7. Виготовити запит з відомостями про студентів. Відмітити відповідним написом порушників, які мали пропуски занять з неповажних причин.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 15 Газосховища України

Задача 1. Спроектувати та утворити базу даних Gazoshov для зберігання та обробки інформації, яка містить такі дані про газосховища України:

- 1– назва комплексу;
- 2– назва газового сховища;
- 3– об’єм газового сховища, млн. м<sup>3</sup>;
- 4– кількість свердловин;
- 5– добова продуктивність свердловин;
- 6– кількість запомпаного газу, млн. м<sup>3</sup>;
- 7– кількість вибраного газу, млн. м<sup>3</sup>;
- 8– дата занесення даних.

Таблиця 2.15 – Газосховища України

1	2	3	4	5	6	7	8
Західно-Український	Луцьке	8150	12	12	19,6	210,4	04.04.1990
Західно-Український	Черепівське	1000	33	19	118,1	39,1	05.06.1990
Західно-Український	Черепівське	1000	33	19	19,2	120,2	07.09.1989
Київський	Тетіївське	4000	147	8	631,3	81,2	23.09.1990
Київський	Тетіївське	4000	147	8	678,7	72,4	12.08.1991
Київський	Таращанське	1800	34	3	91,9	33,2	20.09.1988

Київський	Таращанське	1800	34	3	80,0	34,1	12.11.1990
Київський	Таращанське	1800	34	3	310,5	310,0	23.10.1988
Донецький	Володарське	300	57	5	120,2	42,4	26.09.1991
Донецький	Крапське	2000	66	5	352,5	36,5	17.11.1989
Південно-Український	Каховське	1740	81	6	573,2	387,4	15.12.1991
Південно-Український	Новоліське	1235	55	7	45,0	34,0	12.09.1992
Південно-Український	Новоліське	1235	55	7	64,0	45,0	15.06.1993

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Комплекс: код комплексу, назва комплексу;
- 2– Сховище: код комплексу, код газового сховища, назва газового сховища, об’єм газового сховища, кількість свердловин, добова продуктивність свердловин;
- 3– Облік газу: код газового сховища, кількість запомпаного газу, кількість вибраного газу, дата занесення даних.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості сумарного запомпаного газу по кварталах за минулий рік для кожного сховища. Навпроти назв комплексів і сховищ, розділених пробілом, установити значення кількості за кожний квартал. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про щомісячний вибір газу протягом поточного року для заданого сховища.

Задача 5. Виготовити звіт про ефективність використання об’єму сховищ у поточному році. Ефективність використання об’єму сховища дорівнює сумарній різниці кількості всього запомпаного і вибраного газу, поділеній на об’єм газового сховища. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту всіх таблиць з даними та доповнення БД відомостями про кількість запомпаного та вибраного газу.

Задача 7. Виготовити запит з відомостями про сховища, вказати назву комплексу. Забезпечити відповідний напис для тих сховищ, де сумарний відбір газу в поточному році перевищує його прихід.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 16 Вміст нафтопродуктів у нафтобазах України

Задача 1. Спроектувати та утворити базу даних Naftbaza для зберігання та обробки інформації, яка містить такі дані про вміст нафтопродуктів у нафтобазах України:

- 1– назва регіону;
- 2– назва області;
- 3– назва нафтобази;
- 4– кількість резервуарів;
- 5– об’єм резервуарів, м<sup>3</sup>;
- 6– кількість нафтопродукту, тис. т;
- 7– дата занесення даних.

Таблиця 2.16 – Вміст нафтопродуктів у нафтобазах України

1	2	3	4	5	6	7
Передкарпатська нафтогазоносна область	Івано-Франківська	Снятин НБ	28	93053	188,200	12.08.1997
Передкарпатська нафтогазоносна область	Закарпатська	Майдан НБ	36	9479	80,060	09.06.1995
Передкарпатська нафтогазоносна область	Закарпатська	Чоп НБ	37	16936	77,432	23.08.1991

Передкарпатська нафтогазоносна область	Чернівецька	Вижниця НБ	210	18114	30,400	20.12.1995
Передкарпатська нафтогазоносна область	Чернівецька	Чернівці НБ	16	16902	138,150	23.11.1998
Передкарпатська нафтогазоносна область	Львівська	Ходорів НБ	110	347190	87,641	15.12.1990
Передкарпатська нафтогазоносна область	Львівська	Ходорів НБ	110	347190	0,002	18.06.1990
Передкарпатська нафтогазоносна область	Львівська	Яворів НБ	320	172652	63,380	28.07.1990
Дніпровсько-Донецька нафтогазоносна область	Луганська	Мілове НБ	44	3066	28,700	17.03.1993
Дніпровсько-Донецька нафтогазоносна область	Харківська	Есхар НБ	156	15675	24,711	25.11.1997
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Сарата НБ	40	15610	60,400	15.12.1990
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Сарата НБ	40	15610	44,400	08.03.1992
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Городнє НБ	25	7526	28,800	06.07.1998
Причорноморсько-Кримська газонафтоносна впадина	Одеська	Берізка НБ	333	12755	24,400	02.04.1998
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Бахчи НБ	25	7053	41,300	12.02.1999
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Судак НБ	33	10527	52,440	19.10.1991
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Бахчи НБ	25	7053	49,900	11.12.1991

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - Регіон: код регіону, назва регіону;
- 2 - Область: код регіону, код області, назва області;
- 3 - Нафтобаза: код області, код нафтобази, назва нафтобази, кількість резервуарів, об'єм резервуарів;
- 4 - Облік: код нафтобази, кількість завезених нафтопродуктів, дата занесення даних.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл нафтобаз у областях по кількості резервуарів. Нафтобази розділити на 4 групи з кількістю резервуарів до 100, 101-200, 201-300 і більше. Навпроти назв регіонів і областей, розділених пробілом, установити значення кількості для кожної групи. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про помісячну інтенсивність використання резервуарів заданої нафтобази у поточному році. Інтенсивність використання резервуарів дорівнює сумі кількості завезених нафтопродуктів за місяць, поділеній на об'єм резервуарів.

Задача 5. Виготовити звіт про кількість поступлення нафтопродуктів у кожну нафтобазу в поточному кварталі. Включити назви регіонів і областей. До заголовка звіту додати слова "станом на" і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про поступлення нафтопродуктів.

Задача 7. Виготовити запит з відомостями про нафтобази. Відмітити відповідним написом ті нафтобази, де інтенсивність використання резервуарів більша за одиницю.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 17 Забезпеченість паливно-мастильними матеріалами**

Задача 1. Спроектувати та утворити базу даних Palyvo для зберігання та обробки інформації, яка містить такі дані про забезпеченість паливно-мастильними матеріалами:

- 1– назва регіону;
- 2– назва області;
- 3– назва нафтобази;
- 4– дата представлення даних;
- 5– кількість бензину, тис. т;
- 6– кількість дизпалива, тис. т;
- 7– кількість машинного масла, тис. т.

Таблиця 2.17 – Забезпеченість паливно-мастильними матеріалами

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>
Передкарпатська нафтогазоносна область	Львівська	Броди НБ	09.01.1993	28,800	28,600	1,400
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Гадяч НБ	19.04.1994	57,188	74,348	0,826
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Глобине НБ	23.01.1992	11,540	12,041	0,253
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Піски НБ	24.02.1996	8,918	11,698	0,144
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Кобеляки НБ	09.04.1991	81,000	62,500	9,700
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Котельва НБ	05.05.1997	29,900	32,900	1,400
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Чутове НБ	03.07.1998	32,631	38,440	3,136
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Миргород НБ	09.03.1992	6,490	13,955	1,332
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Алушта НБ	06.04.1999	10,700	18,900	0,900
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Ялта НБ	18.02.1990	51,200	31,200	1,100
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Судак НБ	22.05.1993	43,600	58,500	1,200
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Котельва НБ	06.04.1999	9,070	3,980	8,900
Причорноморсько-Кримська газонафтоносна впадина	Кримська АР	Ялта НБ	03.07.1998	12,678	89,000	0.800
Дніпровсько-Донецька нафтогазоносна область	Полтавська	Миргород НБ	18.02.1990	89,982	45,050	0,765
Передкарпатська нафтогазоносна область	Івано-Франківська	Одаї НБ	09.03.1992	22,087	13,089	0,500

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - Регіон: код регіону, назва регіону;
- 2 - Область: код регіону, код області, назва області;
- 3 - Нафтобаза: код області, код нафтобази, назва нафтобази;

4 - Паливо: код нафтобази, дата представлення даних, кількість бензину, кількість дизпалива, кількість машинного масла.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості бензину і дизпалива у областях по півріччях станом на поточну дату. Навпроти назв регіонів і областей, розділених словами “регіон:”, “область:”, установити значення всіх кількостей для кожного півріччя. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про dennу кількість бензину протягом поточної декади у заданій нафтобазі.

Задача 5. Виготовити звіт про наявність нафтопродуктів у нафтобазах станом на поточну дату. Перед назвою нафтобази додати назву області і скорочену назву регіону. До заголовка звіту додати слова “станом на ” і встановити поточну дату. врахувати, що дані в базу поступають нерегулярно, тому в звіт необхідно внести останні дані (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про наявність нафтопродуктів.

Задача 7. Виготовити запит з відомостями про нафтобази. Встановити відповідний коментар, якщо середньомісячний вміст нафтопродуктів в поточному місяці перевищує 100 тис. т.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 18 Добовий видобуток нафти на Україні**

Задача 1. Спроектувати та утворити базу даних Vudobuv для зберігання та обробки інформації, яка містить такі дані про добовий видобуток нафти на Україні:

- 1– назва нафтогазоносного регіону;
- 2– назва області;
- 3– назва нафтового родовища;
- 4– дата представлення даних;
- 5– адреса родовища;
- 6– рік введення родовища в експлуатацію;
- 7– добовий видобуток нафти, тон;
- 8– кількість свердловин родовища.

Таблиця 2.18 – Добовий видобуток нафти на Україні

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Дніпровсько-Донецький	Херсонська	Каховське-Н	12.09.1995	5 км на ПнС від м. Херсон	1990	6120	12
Дніпровсько-Донецький	Херсонська	Каховське-Н	04.05.1997	5 км на ПнС від м. Херсон	1990	970	12
Дніпровсько-Донецький	Полтавська	Хорольське-НГК	03.08.1995	9 км на ПдС від м. Кобеляки	1964	58766	25
Дніпровсько-Донецький	Дніпропетровська	Могилівське-НГ	17.12.1995	3 км від м. Новомосковськ	1976	4590	43
Дніпровсько-Донецький	Полтавська	Хорольське-НГК	16.10.1995	9 км на ПдС від м. Кобеляки	1964	2250	25
Дніпровсько-Донецький	Дніпропетровська	Роздорівське-НГК	07.12.1995	21 км на ПнЗ від с. Іскра	1985	3050	23
Дніпровсько-Донецький	Сумська	Левадівське-НГК	07.03.1999	100 км на ПнС від м. Ромни	1973	6787	35

Дніпровсько-Донецький	Сумська	Корецьке-НГК	28.05.1995	50 км на ПнЗ від м. Ромни	1990	31575	64
Дніпровсько-Донецький	Сумська	Корецьке-НГК	22.11.1996	50 км на ПнЗ від м. Ромни	1990	84161	64
Передкарпатський	Івано-Франківська	Коломийське-Н	11.12.2000	60 км на П від м. Ів-Франківськ	1956	28459	34
Передкарпатський	Івано-Франківська	Калуське-НГК	23.12.1997	30 км на З від м. Ів-Франківськ	1962	7108	37
Передкарпатський	Львівська	Ходорівське-НГК	29.01.1993	10 км на ПдЗ від м. Львів	1954	59025	45
Передкарпатський	Львівська	Глинянське-Н	28.02.1998	20 км на ПдЗ від м. Бібрка	1967	1934	22
Передкарпатський	Львівська	Стрийське-Н	22.09.1995	15 км на ПдЗ від м. Стрий	1890	4527	18
Передкарпатський	Івано-Франківська	Тлумацьке-Н	17.09.1995	7 км на ПдС від м. Ів-Франківськ	1967	9164	35
Передкарпатський	Івано-Франківська	Тлумацьке-Н	19.07.1997	7 км на ПдС від м. Ів-Франківськ	1967	1784	35

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - Регіон: код нафтогазоносного регіону, назва нафтогазоносного регіону;
- 2 - Область: код регіону, код області, назва області;
- 3 - Родовище: код області, код родовища, назва нафтового родовища, адреса родовища, рік введення родовища в експлуатацію, кількість свердловин;
- 4 - Видобування: код родовища, дата представлення даних, видобуток нафти за добу.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл сумарного видобутку нафти в областях по кварталах з дня відкриття родовищ. Навпроти назв регіонів і областей, розділених пробілом, установити кількість видобутку для кожного кварталу. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку введення родовищ в експлуатацію для заданого регіону з кроком дискретизації періоду введення в розробку, який дорівнює 50 років. Тобто роки введення родовищ у розробку згрупувати, утворити 3 періоди: до 1950 року, 1951-2000 рр., від 2001 р. до поточної дати.

Задача 5. Виготовити звіт про сумарний видобуток нафти в кожному родовищі протягом останнього місяця. Підвести підсумки для областей. До заголовка звіту додати слова “станом на” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про добовий видобуток нафти.

Задача 7. Виготовити запит з відомостями про родовища. Вказати, що родовище молоде, якщо воно введено розробку не пізніше як 20 років тому.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 19 Кадровий облік працівників фірми

Задача 1. Спроектувати та утворити базу даних Kadru для зберігання та обробки інформації, яка містить такі дані про кадровий облік працівників фірми “Візерунки Прикарпаття”:

- 1 – табельний номер працівника;
- 2 – прізвище;
- 3 – ім’я;
- 4 – по-батькові;

- 5– посада;  
 6– освіта;  
 7– дата влаштування на роботу;  
 8– спеціальність.

Таблиця 2.19 – Кадровий облік працівників фірми “Візерунки Прикарпаття”

1	2	3	4	5	6	7	8
000501	Іванів	Василь	Романович	директор	вища	02.03.99	інженер-економіст
000502	Іvasишин	Галина	Петрівна	менеджер	вища	10.12.98	інженер
000503	Василишин	Петро	Іванович	головний інженер	вища	12.10.88	інженер
000504	Васько	Ольга	Петрівна	заступник директора	вища	08.02.89	інженер
000505	Дрінь	Оксана	Іванівна	технолог	вища	02.04.99	інженер
000506	Дудка	Віктор	Павлович	механік	вища	10.10.96	інженер-механік
000507	Світко	Інна	Миколаївна	швея	середня спеціальна	11.07.85	швея-мотористка
000508	Яремич	Віктор	Павлович	електрик	середня спеціальна	25.02.78	швея мотористка
000509	Круць	Іван	Іванович	майстер	вища	14.02.78	інженер механік
000510	Барна	Наталія	Іванівна	швея	середня спеціальна	05.14.78	швея
000511	Брик	Богдан	Михайлович	начальник цеху	вища	12.05.99	інженер
000512	Заремба	Іванна	Іванівна	швея	середня спеціальна	14.02.98	швея
000513	Якимчук	Людмила	Богданівна	швея	середня спеціальна	14.01.76	швея
000514	Якимів	Віра	Олексіївна	швея	середня спеціальна	03.05.87	закрійник
000515	Таран	Оксана	Олексіївна	швея	середня спеціальна	08.10.94	модельєр
000516	Василів	Марія	Петрівна	швея	середня спеціальна	04.05.93	закрійник

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 – Працівники: табельний номер, прізвище, ім’я, по-батькові, код посади, код освіти, дата влаштування на роботу, код спеціальності;
- 2 – Освіта: код освіти, ступінь освіти;
- 3 – Посада: код посади, посада;
- 4 – Спеціальність: код спеціальності, спеціальність.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості посад працівників за ступенем освіти. Навпроти назв посад, які мають починатися з великої букви, установити кількість працівників для кожного ступеня освіти. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит кількості поступлення на роботу працівників кожних 20 років протягом останніх 60 років.

Задача 5. Виготовити звіт про працівників, які після завтра будуть справляти 10-літній ювілей поступлення на роботу. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про нових працівників.

Задача 7. Виготовити запит з відомостями про працівників. Додати напис, що працівник є ветераном праці, якщо його стаж роботи більший, ніж 25 років.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 20 Дані про туристичну фірму (даний варіант розглянути в прикладі)**

Задача 1. Спроектувати та утворити базу даних Turyst для автоматизації послуг фірми “Сьорфінг”, яка містить такі дані про фірму “Сьорфінг”:

- 1– найменування маршруту;
- 2– кількість днів;
- 3– дата початку маршруту;
- 4– поточна кількість проданих путівок, шт;
- 5– умови проживання;
- 6– умови харчування;
- 7– ціна за одну путівку, у. о.;
- 8– початкова кількість путівок, шт.

Таблиця 2.20 – Туристична фірма “Сьорфінг”

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
Австрія (Альпи)	8	01.02.05	8	3* і 5* готелі	3-разове харчування	850	10
Англія (Лондон)	11	05.03.05	15	3* готелі	3-разове харчування	610	20
Гаваї	12	01.02.05	0	3* і 5* готелі	сніданок	565	44
Угорщина (Будапешт)	6	01.02.05	0	3* готелі	сніданок	99	34
Єгипет (Хургада)	7	06.06.05	120	2* і 3* готелі	3-разове харчування	219	126
Польща	7	20.10.05	0	3* і 5* готелі	3-разове харчування	99	224
Польща (Краків)	5	06.06.05	442	3* готелі	сніданок	99	442
Словакія	7	05.03.05	200	3* і 5* готелі	3-разове харчування	255	224
Словакія, Польща	7	01.02.05	0	2* готелі	сніданок	264	100
Тайланд	7	01.04.05	22	3* і 5* готелі	3-разове харчування	700	48
Чехія (Прага)	6	06.06.05	34	3* готелі	сніданок	551	88
Швейцарія	8	01.02.05	47	3* готелі	сніданок	950	124
О. Бали	8	06.06.05	47	2* готелі	сніданок	1190	48
О. Маврикій	8	20.10.05	10	2* і 3* готелі	3-разове харчування	1390	48
Йорданія	7	05.03.05	54	2* готелі	без харчування	330	68
Йорданія	14	20.10.05	12	3* готелі	без харчування	660	48

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - Маршрут: номер маршруту, найменування маршруту, кількість днів, кількість путівок, код умов проживання, код умов харчування, ціна за одну путівку;
- 2 - Облік реалізації: номер маршруту, дата початку маршруту, поточна кількість проданих путівок;
- 3 - Проживання: код умов проживання, умови проживання;
- 4 - Харчування: код умов харчування, умови харчування.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості путівок з певними умовами проживання по умовах харчування. Навпроти умов проживання встановити кількість путівок для кожного виду харчування. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Утворити запит про помісячну сумарну виручку за путівки в поточному році. Виручка за путівки дорівнює її ціні, помножений на кількість проданих.

Задача 5. Виготовити звіт про “гарячі” путівки, тобто такі, що ще незакуплені, а до початку маршруту залишилося менше 6 днів. У звіт включити дані про номер і назву маршруту, ціну за одну путівку та кількість путівок, які ще залишилися. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про нові маршрути і редагування кількості проданих путівок.

Задача 7. Виготовити запитки з відомостями про путівки фірми. Забезпечити відповідний напис для зимових путівок, тобто там, де дата початку маршруту припадає на місяці від грудня до лютого включно.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 21 Продажа товарів торгівельною фірмою**

Задача 1. Спроектувати та утворити базу даних Tovar для зберігання та обробки інформації, яка містить такі дані про продажу товарів фірмою “Какаду”:

- 1– назва виробу;
- 2– постачальник товару;
- 3– дата обліку;
- 4– торгова точка фірми “Какаду”;
- 5– кількість реалізованого товару, шт;
- 6– ціна однієї упаковки виробу, грн;
- 7– наявність 10%-ної націнки за поставку товару.

Таблиця 2.21 – Продажа товарів фірмою “Какаду”

1	2	3	4	5	6	7
Соєва олія	ПП ”Олеся” Запоріжжя	12.06.02	Бар “Асторія”	200	3,25	Так
М’ясо соєве	ПП “Флора” Київ	03.09.02	магазин № 15	400	2,20	Так
Соєвий сир Тофу	Центральна товарна база	04.05.02	Кафе “Капітан”	890	10,15	Ні
М’ясо соєве	Центральна товарна база	23.07.02	Бар “Асторія”	450	2,20	Ні
Соєвий йогурт	ПП “Флора” Київ	18.12.02	магазин № 21	2000	1,80	Так
Соєве борошно	Центральна товарна база	19.04.02	Кафе “Капітан”	370	1,20	Ні
М’ясо соєве	ПП “Флора” Київ	26.01.02	Бар “Асторія”	450	2,20	Так
Соус соєвий	ПП ”Олеся” Запоріжжя	23.07.02	магазин № 12	600	2,25	Так
Соєва олія	ПП ”Олеся” Запоріжжя	04.03.02	магазин № 12	1200	3,25	Так
Соєвий сир Тофу	ПП “Флора” Київ	15.10.02	Кафе “Капітан”	1250	10,15	Так
М’ясо соєве	ПП “Флора” Київ	18.11.02	магазин № 15	2020	2,20	Так
Соєвий сир Тофу	Центральна товарна база	04.02.02	магазин № 21	890	10,15	Ні
Соєвий сир Тофу	ПП “Флора” Київ	08.03.02	магазин № 15	540	10,15	Так
Соєвий йогурт	Центральна товарна база	10.10.02	Кафе “Капітан”	320	1,80	Ні
Соєва олія	ПП ”Олеся” Запоріжжя	09.11.02	магазин № 15	25	3,25	Так

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Товари: код виробу, назва виробу, ціна за одну упаковку;
- 2– Постачальники: код постачальника товару, постачальник товару, наявність 10%-ної націнки за поставку товару;
- 3– Торгова точка: код торгової точки, назва торгової точки;
- 4– Збут: код виробу, код торгової точки, кількість реалізованого товару, код постачальника товару, дата.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл виручки за реалізовані товари по постачальниках у минулому році. Навпроти назв товарів поставити реалізацію для кожного постачальника.

Реалізація одиниці товару дорівнює кількості продажу, помноженій на ціну або на 1,1 ціни, якщо є 10%-на націнка. Передбачити підсумки по вертикалі і по горизонталі. До заголовка звіту додати слова “станом на” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про реалізацію заданого товару за кожний місяць поточного року.

Задача 5. Виготовити звіт про сумарний прибуток фірми за кожний товар у поточному році. Прибуток одиниці товару дорівнює 30% від виручки. Врахувати 10%-ну націнку (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про реалізацію товару.

Задача 7. Виготовити запит з відомостями про товари. Включити дані про назву та ціну товару. Додати напис про те, що товар вигідний, якщо виручка в минулому році перевищувала 8000 грн.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

## Варіант 22 Продажа фірмою виробів із сої

Задача 1. Спроектувати та утворити базу даних Soy для зберігання та обробки інформації, яка містить такі дані про продажу виробів із сої фірмою “Золотий вік”:

- 1– назва виробу;
- 2– постачальник товару;
- 3– дата обліку;
- 4– торгова точка фірми “Золотий вік”;
- 5– кількість реалізованого товару, шт;
- 6– ціна однієї упаковки виробу, грн;
- 7– ознака передоплати постачальнику.

Таблиця 2.22 – Продажа виробів із сої фірмою “Золотий вік”

1	2	3	4	5	6	7
Соєва олія	ПП ”Олеся” Запоріжжя	12.06.02	Бар “Асторія”	200	3,25	Так
М'ясо соєве	ПП “Флора” Київ	03.09.02	магазин № 15	400	2,20	Так
Соєвий сир Тофу	Центральна товарна база	04.05.02	Кафе “Капітан”	890	10,15	Ні
М'ясо соєве	Центральна товарна база	23.07.02	Бар “Асторія”	450	2,20	Ні
Соєвий йогурт	ПП “Флора” Київ	18.12.02	магазин № 21	2000	1,80	Так
Соєве борошно	Центральна товарна база	19.04.02	Кафе “Капітан”	370	1,20	Ні
М'ясо соєве	ПП “Флора” Київ	26.01.02	Бар “Асторія”	450	2,20	Так
Соус соєвий	ПП ”Олеся” Запоріжжя	23.07.02	магазин № 12	600	2,25	Так
Соєва олія	ПП ”Олеся” Запоріжжя	04.03.02	магазин № 12	1200	3,25	Так
Соєвий сир Тофу	ПП “Флора” Київ	15.10.02	Кафе “Капітан”	1250	10,15	Так
М'ясо соєве	ПП “Флора” Київ	18.11.02	магазин № 15	2020	2,20	Так
Соєвий сир Тофу	Центральна товарна база	04.02.02	магазин № 21	890	10,15	Ні
Соєвий сир Тофу	ПП “Флора” Київ	08.03.02	магазин № 15	540	10,15	Так
Соєвий йогурт	Центральна товарна база	10.10.02	Кафе “Капітан”	320	1,80	Ні
Соєва олія	ПП ”Олеся” Запоріжжя	09.11.02	магазин № 15	25	3,25	Так

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Вироби: код виробу, назва виробу, ціна за одну упаковку.
- 2– Постачальники: код постачальника, постачальник товару.
- 3– Торгова точка: код торгової точки, назва торгової точки.
- 4– Реалізація: код виробу, код торгової точки, код постачальника, кількість, дата.
- 5– Передоплата: код виробу, код постачальника, ознака передоплати постачальнику за товар.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл виручки за реалізовані товари по постачальниках у минулому році. Навпроти назв товарів поставити реалізацію для кожного постачальника. Реалізація дорівнює кількості продажу, помноженій на ціну. Передбачити підсумки по вертикалі і по горизонталі. До заголовка звіту додати слова “станом на ” і встановити поточну дату (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про реалізацію (кількість) заданого товару за кожний місяць поточного року. Відібрати лише товари з передоплатою постачальнику.

Задача 5. Виготовити звіт про сумарний прибуток фірми за кожний товар у поточному році. Прибуток дорівнює 20% від виручки (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про реалізацію товару.

Задача 7. Виготовити запит з відомостями про товари. Включити дані про назву та ціну товару. Додати напис про те, що товар користується підвищеним попитом, якщо його сумарна реалізація в поточному році перевищила 10 000 шт.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 23 Виготовлення продукції підприємствами виробничого об'єднання**

Задача 1. Спроектувати та утворити базу даних Robot для зберігання та обробки інформації, яка містить такі дані про виготовлення продукції підприємствами об'єднання “Веста”:

№ п/п;

1– назва філії;

2– адреса філії;

3– телефон керівника філії;

4– кількість інвестицій в минулому році, грн;

5– ціна одиниці продукту, грн;

6– назва продукту;

7– марка продукту;

8– дата обліку;

9– кількість виготовленого продукту.

Таблиця 2.23 – Виготовлення продукції підприємствами об'єднання “Веста”

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
Магнолія	смт. Тура	17499	7000	5	Генератор	ГНТ-865-6а	18.04.13	64
Горизонт	вул. Суха, 5	12345	0	125	Контактор	ЛКк-17	18.04.13	12
Делона	пер. Юта, 82	54321	1000	66	Корпус	К4	18.04.13	8
Віал	вул. Діброва	98765	800	6	Вилка	ДДВ-07-Е3	18.04.13	128
Ресурс	пл. Пік, 14	23456	0	34	Шайба	Ш16	18.04.13	1230
Магнолія	смт. Тура	17499	7000	345	Кронштейн	КтРН-903	19.04.13	125
Горизонт	вул. Суха, 5	12345	0	125	Контактор	ЛКк-17	19.04.13	15
Делона	пер. Юта, 82	54321	1000	66	Корпус	К4	19.04.13	9
Віал	ур. Діброва	98765	800	765	Плита	П504	19.04.13	2
Ресурс	пл. Пік, 14	23456	0	34	Шайба	Ш16	19.04.13	2250
Магнолія	смт. Тура	17499	7000	5	Генератор	ГНТ-865-6а	20.04.13	68
Горизонт	вул. Суха, 5	12345	0	25	Вісь	ДОш-05-06	20.04.13	200
Делона	пер. Юта, 82	54321	1000	66	Корпус	К4	20.04.13	8
Віал	вул. Діброва	98765	800	765	Плита	П504г	20.04.13	2
Ресурс	пл. Пік, 14	23456	0	34	Шайба	Ш16	20.04.13	1500
Магнолія	смт. Тура	17499	7000	5	Генератор	ГНТ-865-6а	21.04.13	64

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Філія: код філії, назва філії, адреса філії, телефон керівника філії, кількість інвестицій;
- 2– Продукти: код продукту, назва продукту, марка продукту, ціна одиниці продукту;
- 3– Виготовлення: код продукту, код філії, дата обліку, кількість виготовленого продукту.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про поквартальний розподіл доходів філій у поточному році. Навпроти назв філій та назв продуктів, розділених пробілами встановити доходи за кожний квартал. Дохід за одиницею продукції дорівнює 20% від ціни, помноженої на кількість. Передбачити підсумки по вертикальні і по горизонтальні (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про кількість виготовленого заданого продукту за кожний місяць поточного року.

Задача 5. Виготовити звіт про виробничу діяльність філіалів фірми у минулому році. До звіту включити назви і адреси філій, розділених пробілами, назви та марки виготовлених продуктів, теж розділених пробілами, і сумарну кількість продуктів (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про щоденне виготовлення продуктів.

Задача 7. Виготовити запит з відомостями про продукти. Внести назву та марку і ціну продукту, назву та адресу філії і об'єм виготовлення за останній місяць. Додати напис, що інвестиції відсутні, якщо в тому філіалі, де продукти виготовлялися, вони дорівнюють нулю.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 24 Малий бізнес**

Задача 1. Спроектувати та утворити базу даних Biznes для зберігання та обробки інформації, яка містить такі дані про малий бізнес Івано-Франківщини:

- 1– номер п/п;
- 2– назва підприємства;
- 3– адреса підприємства;
- 4– дата реєстрації підприємства;
- 5– статутний фонд, тис. грн;
- 6– керівник;
- 7– вид діяльності підприємства;
- 8– кількість робочих місць.

Таблиця 2.24 – Малий бізнес

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>
1	КП “Житло”	вул. Кропного, 1а	06.05.03	34,5	Українець В.О.	Гуртова торгівля	125
2	ВАТ “Будсервіс”	вул. Макухи, 1а	23.08.03	234,8	Білик В.Й.	Ремонт споруд	891
3	ПП “Вітокс ”	вул. Горського, 40	04.07.03	76,9	Сетанюк І.Г.	Метало-вироби	251
4	ПП “Артем”	вул. Августина Волошина, 9	04.10.03	56,4	Марченко В.П.	Вироби з пластмас	329
5	ВАТ “Колорит”	вул. Пуллюя, 2	04.03.02	564,8	Клипич М.М	Швейні вироби	158
6	ТзОВ “Петра”	вул. Бойчука, 20	03.09.03	422,3	Торбяк В.В.	Вироби з пластмас	320
7	ВАТ “Автомаш”	вул. Юності, 23	09.03.04	75,0	Блонський В.К.	Ливарне обладнання	698
8	ВАТ “Івіта”	вул. Юності, 25	09.03.04	100,0	Бабкова О.Р.	Прачальні машини	981
9	КП “Гефест”	вул. Кілевської, 40	02.04.01	78,9	Клочко М.В.	Ковані вироби	253
10	ПП “Дар”	вул. Галицька, 40	04.09.04	786,5	Микитин М.Д.	Хліб, кекс	128
11	ТзОВ “Рокос”	вул. Конського, 24	06.05.04	56,4	Селепій І.М.	Мінеральні води	103
12	ПП “Аркадія”	вул. Героїв, 6	04.09.04	341,3	Косько М.М.	Хліб	89
13	ПП “Кубі”	вул. Микитська, 92	21.03.04	87,3	Кузьмин І.В.	Ковбасні вироби	98

14	ТзОВ “Стела”	вул. Чорного, 144	06.03.02	23,4	Токарчук В.В.	Хліб	69
15	ПП “Скарт”	вул. Атаманюка, 86	04.09.04	56,5	Гура М.М.	Ковбасні вироби	78
16	Коператив “Росіді”	вул. Ушинського, 2	04.09.04	45,0	Мошняга Г.О.	Ковбасні вироби	45

Утворити таблиці БД. Рекомендації до побудови таблиць:

1– Підприємство: код підприємства, назва підприємства, адреса підприємства, дата реєстрації, статутний фонд, керівник підприємства, код діяльності підприємства, кількість робочих місць;

2– Діяльність: код діяльності підприємства, вид діяльності підприємства  
Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості підприємств за видами діяльності по п'ятирічках реєстрації протягом останніх 15 років. Навпроти назв видів діяльності встановити кількість підприємств за кожне п'ятиріччя. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку поквартальної ( сезонної) зміни кількості робочих місць протягом останнього десятиліття. Внести № кварталу і сумарну кількість робочих місць у кожному кварталі за 10 років.

Задача 5. Виготовити звіт про хід реєстрації малих підприємств, статутний фонд яких не перевищує 100 тис грн. У звіт внести дані про назву підприємства, його адресу, дату реєстрації, прізвище та ініціали керівника, вид діяльності та кількість робочих місць (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду всіх даних (крім кодів) про підприємства. Забезпечити доповнення БД відомостями про реєстрацію нових підприємств.

Задача 7. Виготовити запит з відомостями про підприємства. Занести дані про назву підприємства, його адресу, дату реєстрації, прізвище та ініціали керівника, вид діяльності, статутний фонд і кількість робочих місць. Вказати, що підприємство велике, якщо його статутний фонд перевищує 100 тис. грн.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 25 Книжковий магазин

Задача 1. Спроектувати та утворити базу даних kniga для зберігання та обробки інформації, яка містить такі дані про книжковий магазин:

- 1– назва книги;
- 2– автор;
- 3– жанр;
- 4– дата поступлення в магазин;
- 5– кількість проданих книжок;
- 6– ціна 1 книжки, грн;
- 7– дата продажу;
- 8– кількість поступлених книжок.

Таблиця 2.25 – Книжковий магазин

1	2	3	4	5	6	7	8
Золота людина	М. Йокай	класика	06.01.03	25	12	10.02.13	12
500 рецептів	С. Романова	кулінарія	05.02.03	254	8	02.02.12	24
500 страв дешево	Д. Цвек	кулінарія	01.12.02	651	15	12.12.12	0
300 рецептів млинців	Д. Цвек	кулінарія	25.10.03	88	11	25.12.13	25
Мертві яхта	Ч. Вільямс	детектив	14.12.01	54	8	14.12.12	67
По той бій закону	О. Стрендж	детектив	02.04.02	100	9	02.11.12	0

Брат мій, ворог мій	М. Уілсон	пригоди	02.10.01	451	8,5	23.05.09	32
Примари Всесвіту	А. Бороденков	фантастика	05.01.02	12	6	05.01.07	2
Рама II	А. Кларк	фантастика	01.02.03	69	7,5	08.02.13	45
В пошуках Рейчел Уоллес	Р. Паркер	детектив	15.01.03	421	9,6	04.01.13	18
Біс у ребро	М. Серова	пригоди	18.02.02	102	5	18.11.12	22
Естафета смерті	Ж. де Вільс	детектив	19.11.02	200	6	19.12.12	12
Підводний човен, що впав з неба	Д. Соболь	фантастика	14.10.02	25	8	14.12.12	15
Хижак	Дж. Томас	пригоди	18.10.02	123	7	12.06.09	60
Чекаючи на Апокаліпсис	Ч. Абдулаєв	пригоди	02.12.02	89	5	02.02.08	90
Повісті та оповідання	Л. Толстой	класика	10.10.02	20	25	01.01.07	40

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Книга: код книги, код автора, код жанру, назва книги, ціна 1 книжки;
- 2– Автор: код автора, автор;
- 3– Жанр: код жанру, жанр;
- 4– Поступлення: код книги, кількість поступлених книжок, дата поступлення в магазин;
- 5– Реалізація: код книги, кількість проданих книжок, дата продажу.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості книжок кожного жанру по часу перебування в магазині з градацією у 100 днів. Час перебування книжки в магазині дорівнює різниці дати продажу і поступлення. Якщо книга ще не продана, то замість дати продажу поставити поточну дату. Навпроти назв жанрів установити кількість книжок, які перебували в магазині кількість днів: до 100, від 101 до 200 і більше 200. Передбачити підсумки по вертикалі і по горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про кількість книжок заданого жанру, поступлених у кожному місяці поточного року.

Задача 5. Виготовити звіт про хід реалізації книжок у поточному році. Занести назву і ціну книги, автора, жанр і кількість продажу (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць зі всіма даними та доповнення БД відомостями про поступлення нових книг і хід реалізації.

Задача 7. Виготовити запит з відомостями про ще не продані книжки. Внести назву і кількість непроданих книг, автора, жанр, дату поступлення і ціну.

Задача 8. Скласти програму для формування і видачі звіту за умовою задачі 5.

## Варіант 26 Заклади освіти

Задача 1. Спроектувати та утворити базу даних Osvita для зберігання та обробки інформації, яка містить такі дані про заклади освіти Івано-Франківщини:

- 1– номер п/п;
- 2– назва закладу;
- 3– адреса;
- 4– телефон;
- 5– тип закладу;
- 6– керівник освітнього закладу;
- 7– кількість студентів, поступлених у поточному році;
- 8– дата занесення даних.

Таблиця 2.26 – Заклади освіти

1	2	3	4	5	6	7	8
1	Економіко-правничий інститут	вул. Мельничука, 14	72117	ВНЗ	Грита Я.В	850	2.12.12

2	Івано-Франківська державна медична академія	вул. Галицька, 2	24295	ВНЗ	Нейко Є.М.	2560	2.12.12
3	Івано-Франківський Національний технічний університет нафти і газу	вул. Карпатська, 15	42453	ВНЗ	Крижанівський Є.І.	6500	2.12.12
4	Прикарпатський університет ім. В.Стефаника	вул. Шевченка, 55	31574	ВНЗ	Кононенко І.М.	3260	2.12.12
5	Івано-Франківський інститут менеджменту	вул. Дністровська, 28	32464	ВНЗ	Бакаляр Г.Н.	1250	2.12.12
6	ВХПУ №3	вул. Дудаєва, 35	32456	ПТУ	Данилюк Б.П.	450	2.12.12
7	ВХПУ №5	вул. Юності, 13	22429	ПТУ	Кухній М.М.	280	2.12.12
8	ВХПУ №13	вул. Пасічна, 10а	55051		Кінаш І.В.	125	2.12.12
9	ВХПУ №15	вул. Юності, 3	24762	ПТУ	Гуцул І.С.	180	2.12.12
10	Івано-Франківське базове медичне училище	вул. Мазепи, 165	35222	СПНЗ	Стасюк В.В	250	2.12.12
11	Івано-Франківське музичне училище	вул. Січових Стрілців, 36	22365	СПНЗ	Мокрецький М.С.	240	2.12.12
12	Івано-Франківський коледж фізичного виховання	вул. Мазепи, 142	31192	СПНЗ	Турковський Я.О	380	2.12.12
13	Івано-Франківський комерційний коледж	вул. Степана Бандери, 79	25543	СПНЗ	Кафлан В.М.	450	2.12.12
14	Коледж електронних приладів	вул. Вовченецька, 223	67811	СПНЗ	Кельбас Б.А	850	2.12.12
15	Івано-Франківський сільськогосподарський технікум	вул. Юності, 11	22695	СПНЗ	Мотрук Б.Н.	640	2.12.12

Скорочення: СПНЗ – Середньо-професійний навчальний заклад; ПТУ – Професійно-технічне училище; ВНЗ – Вищий навчальний заклад

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - Заклад: код закладу, назва закладу, код типу закладу, керівник закладу, адреса закладу, номер телефону;
- 2 - Типи закладів: код типу закладу, тип закладу;
- 3 - Збір інформації: код закладу, кількість студентів, дата збору інформації.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості студентів кожного року по типах закладів (в рядках – роки, у стовпчиках – типи закладів) з дня заснування закладу, підвести загальний підсумок за всі роки (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про сумарну кількість студентів для середньо-професійних типів закладів за кожний рік протягом останніх 5 років.

Задача 5. Виготовити звіт про наявність студентів у кожному закладі станом на поточну дату. Врахувати, що термін навчання у вищих навчальних закладах дорівнює 5 років, решта – 3 роки. Підвести підсумок по області (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про кількість студентів.

Задача 7. Виготовити запит з відомостями про заклади освіти, в якій вказати назву закладу, адресу, тип закладу, керівника освітнього закладу та середньорічну кількість студентів за весь період обліку.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

## Варіант 27 Науково-технічна література бібліотеки

**Задача 1.** Спроектувати та утворити базу даних Book для зберігання та обробки інформації, яка містить такі дані про науково – технічну літературу бібліотеки:

- 1– номер книги;
- 2– назва книги;
- 3– автор;
- 4– напрямок;
- 5– кількість книжок у бібліотеци;
- 6– кількість сторінок;
- 7– дата поступлення в бібліотеку;
- 8– видавництво.

Таблиця 2.27 – Науково – технічна література бібліотеки

1	2	3	4	5	6	7	8
1	Економічний аналіз	Чумachenko M.G	економіка	48	232	21.01.95	КНЕУ Київ
2	Контроль і ревізія	Романів Є.М.	економіка	12	320	22.07.05	Львів “Інтелект-Захід”
3	Фінансовий облік	Пушкар М.С.	економіка	10	628	23.04.02	Тернопіль “Факля”
4	Гроші. Фінанси. Кредит.	Петровська І.О	економіка	27	300	01.02.02	КНЕУ Київ
5	Економіка підприємства	Данилюк М.О	економіка	125	320	04.12.98	Івано-Франківськ ISBN
6	Економіка підприємства	Примак Т.О.	економіка	10	176	22.04.05	Івано-Франківськ ISBN
7	Сучасні ділові папери	Глушник С.В.	мовознавство	87	539	23.06.00	Івано-Франківськ ISBN
8	Латинська мова	Оленич Р.М.	мовознавство	49	352	30.03.04	Львів Світ
9	Словник українських морфем	Палюга Л. М.	мовознавство	74	448	31.01.00	Львів Світ
10	Штучний інтелект	Глибовець М.М.	комп’ютерні науки	148	366	12.02.89	Івано-Франківськ ISBN
11	Комп’ютерні мережі	Маслов І.В.	комп’ютерні науки	27	38	23.11.08	ІФДТУНГ
12	Бази даних	Глушаков С.В.	комп’ютерні науки	145	304	22.04.00	КНЕУ Київ
13	Програмування на Visual Basic	Глушаков С.В.	комп’ютерні науки	14	498	22.04.05	ФОЛІО Харків
14	Програмування на Visual C++	Глушаков С.В., Коваль А.В.	комп’ютерні науки	28	726	12.02.12	ФОЛІО Харків
15	Програмування на Delphi	Глушаков С.В., Клівцов А.Л	комп’ютерні науки	24	519	22.04.11	ФОЛІО Харків
16	Мова програмування C++	Ліппман С.С	комп’ютерні науки	34	114	12.02.09	БІНОМ Москва

Утворити таблиці БД. Рекомендації до побудови таблиць:

Напрям: код напряму, напрям;

1– Автор: код автора, автор та співавтори;

2– Облік книг у бібліотеци: код книги, назва книги, кількість сторінок, код напряму, код автора, код видавництва, кількість книжок, дата поступлення в бібліотеку;

3– Видавництво: код видавництва, видавництво.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл сумарної кількості книжок видавництв по напрямках (у рядках – назви видавництв, у стовпчиках – напрямки) (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про сумарну кількість поступлених книжок у кожному кварталі минулого року.

Задача 5. Виготовити звіт про недостатню кількість літератури в бібліотеці станом на поточну дату. У звіті передбачити такі відомості: назву книги, автори, напрямок, видавництво, кількість і дату поступлення. Занести у звіт ті книги, сумарна кількість яких менша за 1000 штук і які поступили до бібліотеки 10 і більше років тому. Підвести загальний підсумок (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними про книги та доповнення БД відомостями про поступлення нових книг.

Задача 7. Виготовити запит зі всіма відомостями про книги, поступлені в поточному році.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 28 Транспортні підприємства

Задача 1. Спроектувати та утворити базу даних Transport для зберігання та обробки інформації, яка містить такі дані про транспортні підприємства Івано-Франківщини:

- 1– № підприємства;
- 2– назва підприємства;
- 3– адреса;
- 4– дата заснування підприємства;
- 5– статутний фонд, млн. грн;
- 6– балансова вартість основних фондів, млн. грн;
- 7– обсяг продукції у поточному році, млн. грн;
- 8– кредиторська заборгованість, грн;
- 9– частка акцій трудового колективу, %;
- 10– зовнішні акціонери, %.

Таблиця 2.28 – Транспортні підприємства

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
1	БАТ “Івано-Франківська автобаза”	м. Івано-Франківськ	04.02.96	147,3	174,5	39,5	1268,5	30,1	69,9
2	БАТ “Автопарк”	м. Надвірна	02.07.98	378	451,1	11	114623	84,6	0,0
3	БАТ “Коломийське автотранспортне підприємство”	м. Коломия	24.09.97	660	94	76	0	51,3	18,7
4	БАТ “М’ясомолтранс”	м. Івано-Франківськ	25.03.00	803	131	30	332438	32,4	67,6
5	БАТ “Надвірнянське АТП 12655”	м. Надвірна	30.05.98	29	573	77,7	0	37,4	62,6
6	БАТ “Рогатинавто”	м. Рогатин	02.11.98	26	13,1	20	4872,7	70,2	0,0
7	БАТ “Долинське АТП 12627”	м. Долина	22.03.96	417	463	137	1196	88,7	11,3
8	БАТ “Долинаавто”	м. Долина	27.02.95	285	476	76	0	35,1	34,9
9	БАТ “Городенкаавто”	м. Городенка	22.02.98	160	296,3	185	232662	87,7	0,0
10	БАТ “Калуська автобаза”	м. Калуш	23.08.00	21,2	10	101	0	27,7	72,3
11	БАТ “Івано-Франківськбудтранс”	м. Івано-Франківськ	31.12.02	14,3	15	22	0	41,4	58,6
12	БАТ “Автомобіліст”	м. Калуш	23.01.99	37,08	57	31	3357	66,7	33,3
13	БАТ “Калуське автотранспортне підприємство”	м. Калуш	02.03.01	85	14	12	0	45,7	44,3
14	БАТ ”Івано-Франківське АТП 0927”	м. Івано-Франківськ	03.07.95	362	224	253	0	64,3	0,0
15	БАТ “Івано-Франківський шиноремонтний завод”	м. Івано-Франківськ	14.11.95	313	73	184	0	16,7	37,7
16	БАТ ”Автотранс”	м. Калуш	12.12.02	320	555	722	7981	95,6	0,0

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Підприємства: код підприємства, назва підприємства, адреса, дата заснування, статутний фонд, балансова вартість основних фондів;
- 2– Показники підприємства: код підприємства, обсяг продукції у поточному році, кредиторська заборгованість, частка акцій трудового колективу, зовнішні акціонери.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості підприємств в населених пунктах по балансових вартостях основних фондів з градацією 200 млн. грн (в рядках – назви населених пунктів, у стовпчиках – кількість: до 200, від 201 до 400, від 401 до 600 і т.д.) (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку реєстрації підприємств за останні 5 років – всього і тих, у яких статутний фонд перевищує 100 млн. грн

Задача 5. Виготовити звіт про державну частку акцій у підприємствах області станом на поточну дату. У рядках занести назву підприємства, адресу і статутний фонд, розділені пробілами, у стовпчику – державну частку акцій у відсотках. Підприємства, які не мають державної частки акцій, у звіт не вносити. Державна частка акцій дорівнює 100 мінус (частка акцій трудового колективу + частка акцій зовнішніх акціонерів). Підсумувати кількість підприємств (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запит для перегляду вмісту таблиць з даними та доповнення БД відомостями про реєстрацію нових підприємств.

Задача 7. Виготовити запит зі всіма відомостями про підприємства м. Калуш.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### **Варіант 29 Автомобілі**

Задача 1. Спроектувати та утворити базу даних Avto для зберігання та обробки інформації, яка містить такі дані про автомобілі:

- 1– № п/п;
- 2– прізвище власника;
- 3– адреса власника;
- 4– державний реєстраційний номер;
- 5– марка автомобіля;
- 6– рік випуску;
- 7– дата реєстрації;
- 8– колір автомобіля;
- 9– ціна, у.о.;
- 10– вид пільги у власника, %.

Таблиця 2.29 – Автомобілі

<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>
1	Петренко П.П.	вул. Зелена, 6	450-03-63	Skoda Fabia Basic	1992	01.02.94	червоний	7677	10
2	Руденко І.І.	пер. Лісний, 22	442-40-73	Skoda Octavia	1996	12.12.98	зелений	10650	25
3	Сіренко С.С.	пл. Героїв, 7	553-57-35	Toyota	1998	01.03.99	синій	15700	
4	Дрозд О.О.	вул. Довга, 16	24313-24	Toyota RAV-4	1998	05.05.99	жовтий	24900	
5	Степанов С.К.	вул. Зелена, 12/2	621-51-61	Volvo 440	1989	06.03.90	білий	8250	25
6	Півник П.Р.	пл. Центр, 2	171-35-55	BAZ-21047	2000	12.02.01	чорний	2145	
7	Сіренко С.С.	пл. Героїв, 7	201-49-38	BAZ-21102	1999	12.12.00	сірий	3135	
8	Жаров Д.Т.	вул. Хіміків, 1	201-49-38	BAZ-21099	1998	13.05.99	білий	2887	
9	Ватаг К.Н.	вул. Юр, 34	206-52-56	Mazda Xedos-6	1993	28.07.94	чорний	9000	50

10	Сіренко С.С.	пл. Героїв, 7	502-44-01	Opel Tigra	1990	12.09.92	зелений	6000	
11	Сіренко С.С.	пл. Героїв, 7	422-06-64	Scoda Super	2001	18.09.02	сірий	108287	
12	Арко А.Р.	вул. Денюка, 45	796-04-07	Ford	1990	07.07.91	голубий	2000	10
13	Петренко І.І.	вул. Зелена, 6	789-01-05	Ford	1991	15.07.92	голубий	2500	10
14	Грач Г.Н.	вул. Вільна, 55	243-04-09	Scoda Fabia	1993	08.08.95	чорний	8400	
15	Руденко І.І.	пер. Лісний, 22	512-12-51	Scoda Fabia	1994	12.11.96	білий	8500	25
16	Ранов Р.Р.	вул. Річна, 77	553-57-35	ВАЗ-21102	2000	10.10.01	червоний	5500	

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Автомобілі: код автомобіля, марка автомобіля;
- 2– Власники: код власника, прізвище власника, адреса власника, вид пільги у власника, %;
- 3– Властивості автомобіля: код властивості, рік випуску, ціна в у. о.;
- 4– Колір: код кольору, колір автомобіля;
- 5– Відомості про реєстрацію: код власника, код властивості, державний реєстраційний номер автомобіля, дата реєстрації, код кольору, код автомобіля.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл марок автомобілів по пільгах власників станом на поточну дату. Якщо пільга відсутня, її в звіт не вносити. В рядках відобразити марку автомобіля, в стовпчиках – вид пільги. Підвести підсумки по вертикалі і горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку річного прибутку організації, яка проводить реєстрацію автомобілів, протягом останніх 5 років. Прибуток дорівнює 20% від вартості автомобіля. Власник, який має пільгу, прибутку не дає, у звіт такі відомості не вносити.

Задача 5. Виготовити звіт про наявність автомобілів у власників. Відобразити такі дані: прізвище власника, адреса, вид пільги і кількість автомобілів (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про реєстрацію автомобілів.

Задача 7. Виготовити запит з відомостями про всі зареєстровані автомобілі: марка, рік випуску, ціна, колір.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### Варіант 30 Підприємства-банкрути

Задача 1. Спроектувати та утворити базу даних Bancrut для зберігання та обробки інформації, яка містить такі дані про підприємства-банкрути:

- 1– назва підприємства;
- 2– адреса;
- 3– шифр підприємства;
- 4– № справи;
- 5– дата порушення;
- 6– введена процедура банкрутства;
- 7– кредиторська заборгованість, грн;
- 8– обігові кошти, грн;

Таблиця 3.30 – Підприємства-банкрути

1	2	3	4	5	6	7	8
Кол. буд. компанія “Прибужжя”	м. Хмельницький	375110	13/2-Б	11.12.02	ліквідація	20000,0	15600,0

ВАТ “Хмельницьк-облпром”	м. Хмельницький	125645	4/1-Б	18.12.02	санація	12681,5	25691,0
КП ”Світанок”	м. Полонне	128459	4/78-Б	17.12.01	ліквідація	14623,0	32569,0
Дунаєвецький хлібокомбінат	смт. Дунаївці	102506	2/70-Б	26.11.01	санація	15004,0	48951,0
Ферм. госп-во Желіцина Ю.І.	Летичівський р-н, с. Горбасів	135690	12/68-Б	12.11.01	санація	2438,0	7891,0
ТзОВ ”Регіон”	м. Дунаївці, вул. Шевченка, 98	147890	8/61-Б	08.11.02	ліквідація	1542,8	2569,0
ВАТ “Оргон”	смт. Стара Синява	879120	13/36-Б	03.05.01	санація	4569,2	6897,0
ВАТ “Фарфор”	м. Полонне	789111	2/66-Б	06.11.01	санація	1789,9	2489,0
ВАТ “Керамік”	м. Волочинськ, вул. Короленка, 5	305128	4/62-Б	15.10.01	ліквідація	14569,0	38945,0
Завод “Продтовари”	м. Кам'янець-Подільський	789456	8/51-Б	08.08.01	ліквідація	15890,5	2569,9
Завод “Український кристал”	м. Кам'янець-Подільський	789410	8/52-Б	08.08.01	ліквідація	12369,2	32561,9
ВАТ ”Картон”	м. Полонне	111205	50-Б	27.07.02	ліквідація	74569,0	4567,9

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1– Підприємства: шифр підприємства, назва підприємства, адреса, обігові кошти;
- 2– Відомості про банкрутство: шифр підприємства, кредиторська заборгованість, № справи, дата порушення, код процедури банкрутства;
- 3– Процедура банкрутства: код процедури банкрутства, процедура банкрутства.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити звіт про розподіл кількості адрес підприємств по видах банкрутств станом на поточну дату. В рядках відобразити адреси, у стовпчиках – вид процедури банкрутства, навпротиожної адреси – кількість підприємств. Підвести підсумки по вертикалі і горизонталі (з використанням pl/pgSQL та SQL).

Задача 4. Побудувати запит про динаміку порушень справ про банкрутства за останні 10 років. Внести роки і кількість порушених справ.

Задача 5. Виготовити звіт про банкрутства у поточному році станом на поточну дату. У звіт внести інформацію про назву підприємства, адресу, обігові кошти, кредиторську заборгованість, вид процедури банкрутства (з використанням pl/pgSQL та SQL).

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про нові підприємства.

Задача 7. Виготовити запит зі всіма відомостями про підприємства, кредитна заборгованість яких перевищує 10000 грн.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### 3 ПРИКЛАД ВИКОНАННЯ КУРСОВОЇ РОБОТИ

В даному розділі методичних вказівок показано приклад виконання курсової роботи з дисципліни “Бази даних” (варіант №20).

**КУРСОВА РОБОТА**

**КР. ІП - 25.00.00.000 ПЗ**

Група ІП-19-1

Петров Володимир

2020

Івано-Франківський національний технічний університет нафти і газу  
 (повне найменування вищого навчального закладу)

Кафедра інженерії програмного забезпечення  
 (повна назва кафедри)

## **КУРСОВА РОБОТА**

з дисципліни «Бази даних»

на тему: Реалізація бази даних “Дані про туристичну фірму”  
засобами СКБД PostgreSQL

Студента (ки) 2 курсу групи ІП-19-1  
спеціальності 121 - інженерія  
програмного забезпечення

Петрова В.В.

(прізвище та ініціали)

Керівник

доцент, к.т.н., Вовк Р.Б.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Національнашкала

Кількість балів: \_\_\_\_\_ Оцінка: ECTS \_\_\_\_\_

Члени комісії

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

(підпис)

(прізвище та ініціали)

м. Івано-Франківськ - 2020 р.

Івано-Франківський національний технічний університет нафти і газу

**Кафедра** інженерії програмного забезпечення

**Дисципліна** бази даних

**Спеціальність** 121 – Інженерія програмного забезпечення

**Курс** 2      **Група** ІІІ-19-1      **Семестр** 3

### ЗАВДАННЯ

### НА КУРСОВУ РОБОТУ

**Студенту** Петрову Володимиру Васильовичу

**1. Тема роботи** Реалізація бази даних “Дані про туристичну фірму”

засобами СКБД PostgreSQL

**2. Термін здачі студентом роботи** 01.12.2020

**3. Вихідні дані до роботи** Завдання згідно варіанту № 20

**4. Зміст пояснівальної записки (перелік питань, що їх належить розробити)**

Вступ, теоретична частина (основні відомості про СКБД PostgreSQL),  
практична частина (проектування структури бази даних, створення таблиць та  
обмежень, внесення контрольних даних, побудова запитів, створення програми з  
використанням курсору), висновки, список використаної літератури

**5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)**

---



---



---

**6. Дата видачі завдання** 17 вересня 2020 р.

## КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів	Термін виконання етапів роботи	Об'єм роботи (%)	Відмітка про виконання
1	Одержання завдання, ознайомлення з ним, вибір літератури	17.09.20	—	виконано
2	Вивчення літератури, оформлення теоретичного розділу	23.09.20	10	виконано
3	Розробка структури бази даних	01.10.20	5	виконано
4	Занесення контрольних даних в таблиці	11.10.20	15	виконано
5	Виготовлення запитів для перегляду таблиць з даними	21.10.20	25	виконано
6	Налагодження програми, розв'язування задачі на ПЕОМ	01.11.20	25	виконано
7	Оформлення пояснівальної записки і практичної частини	16.11.20	15	виконано
8	Підготовка роботи до захисту	30.11.20	5	виконано

Студент \_\_\_\_\_  
 (підпис)

Петров В.В.  
 розшифровка підпису

Керівник \_\_\_\_\_  
 (підпис)

Вовк Р.Б.  
 розшифровка підпису

“ 17 ” вересня 2020 р.

## ЗМІСТ

ВСТУП.....	5
2 КОРОТКА ІСТОРІЯ POSTGRESQL.....	6
2.1 Проект POSTGRES, його заснування та розробка.....	6
2.2 Postgres95.....	6
2.3 PostgreSQL.....	7
3 ОСНОВНІ КОНЦЕПЦІЇ РОБОТИ З POSTGRESQL.....	8
3.1 Створення таблиць.....	8
3.2 Внесення даних у таблицю.....	10
3.3 Редагування таблиць.....	11
3.4 Обмеження.....	16
3.5 Послідовності.....	20
3.6 Запити.....	22
3.7 Об'єднання таблиць.....	26
3.8 Агрегатні функції.....	28
3.9 Курсори SQL.....	28
4. ОСНОВНІ ЕЛЕМЕНТИ МОВИ PLpgsql.....	32
4.1 Структура мови, функції користувача.....	32
4.2 Розгалуження, цикли, масиви.....	33
4.3 Курсори.....	35
5 ПРАКТИЧНА ЧАСТИНА.....	37
5.1 Завдання 1 (Створення таблиць, обмежень) .....	38
5.2 Завдання 2 (Заповнення таблиць даними).....	39
5.3 Завдання 4 (Запити).....	41
5.4 Завдання 6 (Запити).....	42
5.5 Завдання 7 (Запити).....	43
5.6 Завдання 3 (Звіти, мова PLpgsql).....	43
5.7 Завдання 5,8 (Звіти, мова PLpgsql).....	45
6. ВИСНОВКИ.....	46
7. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	47

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ		
Розроб.	Петров В.В.				Реалізація бази даних засобами СКБД PostgreSQL	Літ.	Арк.
Перевір.	Вовк Р.Б.					4	Акрушів
Реценз.					Пояснювальна записка		
Н. Контр.					ІФНТУНГ, П-19-1		
Затверд.							

## 1 ВСТУП

PostgreSQL - це об'єктно-реляційна система управління базами даних (ORDBMS) заснована на POSTGRES версії 4.2 , яка була розроблена в Науковому Комп'ютерному Департаменті Берклі Каліфорнійського Університету. Проект POSTGRES, під керівництвом професора Майкла Стоунбрейкера (Michael Stonebraker) був підтриманий Агентством розширення оборонної Дослідних Проектів (Defense Advanced Research Projects Agency (DARPA)), Офісом Армійських Досліджень (Army Research Office (ARO)), Національним Науковим Фондом (National Science Foundation (NSF)), а також ESL, Inc. PostgreSQL є прямим нащадком з відкритим вихідним кодом від оригінального коду, зробленого в Берклі. СУБД надає підтримку SQL92/SQL99 та інші сучасні можливості. POSTGRES є піонером у багатьох об'єктно-реляційних аспектах, що з'явилися тепер в деяких комерційних СУБД. Традиційні реляційні СУБД (RDBMS) підтримують модель даних, яка становить колекцію пойменованих кортежів, що містять атрибути заданого типу. У сучасних комерційних системах, до можливих типів відносяться числа з плаваючою точкою, цілі числа, символні рядки, грошові типи та дати. Це зазвичай призводить до того, що дана модель є неадекватною для майбутніх програм обробки даних. Реляційна модель успішно замінює попередні моделі почасти в силу "спартанської простоти". Однак, така простота робить реалізацію деяких додатків дуже важкою. PostgreSQL пропонує суттєве збільшення потужності СУБД, через впровадження наступних додаткових аспектів, які дозволяють користувачам легко розширювати систему:

- успадкування
- типи даних
- функції

PostgreSQL належить до категорії СУБД, відомих як *об'єктно-реляційні (object-relation)*. Зауважимо, що тут є відмінність від тих *об'єктно-орієнтованих (object-oriented)* СУБД, які в основному підтримують традиційні мови реляційних СУБД. Однак, PostgreSQL має деякі об'єктно-орієнтовані можливості, це важливо в світі реляційних СУБД.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						5

## 2 КОРОТКА ІСТОРІЯ POSTGRESQL

Об'єктно-реляційна СУБД PostgreSQL (стара назва Postgres95) веде своє походження від пакета POSTGRES, який був написаний в департаменті Берклі, Каліфорнійського Університету. Більш ніж десятирічна розробка PostgreSQL зробила цей продукт однією з найбільш потужних СУБД з відкритим вихідним кодом у світі, пропонуючи багатоверсійність управління паралельним доступом, підтримуючи практично всі конструкції SQL (включаючи підзапити, транзакції і визначені користувачем типи та функції) і маючи широкий вибір мов, з допомогою яких можна працювати з СУБД (включаючи C, C++, Java, Perl, Tcl і Python).

### 2.1 Проект POSTGRES його заснування та розробка

Реалізація реляційної СУБД postgres почалася в 1986. Початкові концепції для цієї системи були представлені в “*The design of POSTGRES*”, а визначення початкової моделі даних було здійснено в “*The POSTGRES data model*”. Потім вийшло кілька версій Postgres. Перша "demoware" система запрацювала в 1987 і була продемонстрована в 1988 на Конференції ACM-SIGMOD. Версія 1 була випущена в червні 1989 року і могла працювати з декількома зовнішніми користувачами. Версія 2 випущена в червні 1990 року була заснована на новій системі управління. Версія 3 випущена в 1991, включала в себе підтримку кількох менеджерів зберігання, покращений обробник запитів і мала нову систему управління. Більшість наступних версій до появи Postgres95 були сфокусовані на питаннях переносимості та стабільності. POSTGRES був використаний для реалізації багатьох різних досліджень і написання програм. POSTGRES також використовувався як засіб навчання в декількох університетах. Нарешті компанія Illustra Information Technologies (пізніше стала підрозділом компанії Informix, якою тепер володіє IBM) взяла код цієї СУБД і комерціалізувала його. POSTGRES став пріоритетним менеджером даних для проекту наукових обчислень Sequoia 2000 після 1992 року.

### 2.2 Postgres95

У 1994 році Ендрю Ю і Джоллі Чен додали в POSTGRES інтерпретатор мови SQL. Потім Postgres95 був викладений в Інтернет, як програмний продукт з відкритим вихідним кодом, заснований на оригінальному коді Берклі POSTGRES. Postgres95 був повністю приведений до стандарту ANSI С. Були внесено багато внутрішні зміни, які збільшили продуктивність і налагоджуваність коду. Були зроблені наступні розширення:

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						6

- Підзапити не підтримувалися аж до виходу PostgreSQL, але в Postgres95 їх можна було зімітувати за допомогою функцій SQL, що визначаються користувачем.

- Агрегати були переписані. Також в запиті була додана підтримка GROUP BY. Інтерфейс libpq залишився доступним для програм на C.

- В додаток до програми monitor, була надана нова програма (psql), яка використовувала бібліотеку GNU Readline і була призначена для інтерактивних SQL запитів.

- Створена нова front-end бібліотека, libpgtcl, що підтримує клієнтів, на основі Tcl. Проста оболонка pgsqlsh, що надає нові команди Tcl для забезпечення взаємодії Tcl програм і Postgres95.

- Була ретельно переглянута робота з великими об'єктами. Інверсійні великі об'єкти представляли собою тільки механізм для зберігання великих об'єктів.

- Разом з вихідним кодом став поставлятися короткий підручник з особливостей роботи з SQL вPostgres95.

- Для побудови проекту став використовуватися GNU make. Також, Postgres95 був скомпільзований зі стандартною версією GCC.

## 2.3 PostgreSQL

У 1996 році було вирішено, що ім'я "Postgres95" не відповідає сьогоденню. Було вибрано нове ім'я PostgreSQL щоб підкреслити відмінність від оригінального POSTGRES і вихід багатьох версій з підтримкою SQL.

При розробці Postgres95 акцент ставився на виявлення і розуміння існуючих проблем в коді продукту. В PostgreSQL акцент змістився на розширення можливостей та сумісності при продовженні роботи в усіх інших областях. Головні зміни в PostgreSQL включають:

- Блокування на рівні таблиць була замінена на багатоверсійність управління паралельним доступом, що дозволяє клієнтам здійснювати читання даних під час роботи клієнтів, а також дозволяє здійснювати резервне копіювання програмою pg\_dump в той час, як база залишається доступною для запитів.

- Були реалізовані такі можливості як підзапити, значення по замовчуванню і тригери.

- Були додані можливості для сумісності зі стандартом SQL92, включаючи первинні ключі, ідентифікатори запитів, створення типів, а також двійкове та шістнадцяткове введення цілих чисел.

- Були покращені вбудовані типи даних, включаючи нові широкодіапазонні типи дати / часу і додаткові геометричні типи даних.

- Швидкість роботи backend коду була збільшена приблизно на 20-40%, а час запуску backend'a було скорочено на 80% в порівнянні з версією 6.0.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						7

## 3 ОСНОВНІ КОНЦЕПЦІЇ РОБОТИ З POSTGRESQL

### 3.1 Створення таблиць

PostgreSQL - це система управління реляційними базами даних (СКБД). Це означає, що це система для управління даними, які зберігаються у вигляді відношень (кортежів). Поняття зберігання даних в таблицях є сьогодні таким банальним, що воно може здатися самоочевидним, однак є кілька інших способів організації баз даних. Файли і каталоги в Unix-подібних операційних системах є прикладом ієрархічної бази даних. Одне з найбільш сучасних напрямків розробки СУБД - це об'єктно-орієнтовані бази даних.

Кожна таблиця є пойменованої колекцією *рядків* (*rows*) . Кожен запис у таблиці має деякий набір пойменованих *колонок* (*columns*) і кожне поле є певним типом даних. Поля в записі розташовані в фіксований порядку, а SQL ніколи не гарантує упорядкованого проходження записів в таблиці.

Таблиці групуються в бази даних, а колекція баз даних, керована однією копією сервера PostgreSQL називається *клusterом* баз даних. Команда створення таблиці:

```
CREATE [ TEMPORARY | TEMP ] TABLE ім'я_таблиці (
{ ім'я_поля тип [ обмеження_поля [...] ] | обмеження_таблиці } )
[ INHERITS ( базова_таблиця [ , ... ] ) ] ;
```

де

- **TEMPORARY | TEMP.** Ознака тимчасової таблиці. Таблиця, створена з ключовим словом TEMPORARY або TEMP автоматично знищується наприкінці поточного сесії. Всі конструкції рівня таблиці (наприклад, індекси і обмеження) знищуються наприкінці сесії разом з таблицею. Якщо ім'я тимчасової таблиці співпадає з ім'ям існуючої таблиці, то всі посилання на таблицю з цим ім'ям впродовж сесії відноситимуться до тимчасової таблиці. Іноді це викликає проблеми, оскільки тимчасова таблиця побічно заміщає існуючу таблицю в контексті поточного сесії до моменту її знищення.
- **таблиця.** Ім'я створюваної таблиці.
- **поле.** Ім'я поля в новій таблиці. Імена полів перераховуються в круглих дужках і розділяються комами.
- **тип.** Відразу ж після імені користувача задається його тип – стандартний тип або масив одного із стандартних типів.
- **обмеження\_поля.**
- **базова\_таблиця.** Ім'я таблиці, від якої нова таблиця успадковує поля. Якщо імена успадкованих полів збігаються з іменами полів, раніше включених в структуру таблиці, PostgreSQL видає повідомлення про помилку і перериває виконання команди.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						8

## Типи даних

### Числові типи

smallint	коротке 2-х байтовое ціле
integer	звичайне 4-х байтовое ціле
bigint	велике 8-байтовое ціле
decimal	коротке із фіксованою точкою
numeric	коротке із фіксованою точкою
real	коротке із рухомою точкою
double precision	коротке із рухомою точкою подвійної точності
serial	ціле із автозбільшенням
bigserial	велике ціле із автозбільшенням

### Грошові типи

money	гроші
-------	-------

### Символьні типи

character varying(n), varchar(n)	рядок змінної довжини
character(n), char(n)	рядок фіксованої довжини
text	рядок змінної необмеженої довжини

### Бінарні типи

bytea	бінарний рядок змінної довжини
-------	--------------------------------

### Дата і час

timestamp [ (p) ] [ без часового пояса ]	дата і час
timestamp [ (p) ] с часовим поясом	дата і час із часовим поясом
interval [ (p) ]	Інтервал часу
date	тільки дата
time [ (p) ] [ без часового пояса ]	тільки час
time [ (p) ] с часовим поясом	тільки час із часовим поясом

### Логічні типи

boolean	TRUE або FALSE
---------	----------------

### Геометричні типи

point	Точка на площині (x,y)
line	Невидима лінія (не повністю реалізовано)
lseg	Видимий відрізок ((x1,y1),(x2,y2))
box	Чотирикутник ((x1,y1),(x2,y2))
path	Замкнений прямокутник ((x1,y1),...)
path	Ламана лінія [(x1,y1),...]

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						9

<code>polygon</code>	Полігон ((x1,y1),...)
<code>circle</code>	Круг (x,y),r (центр і радіус)

#### Типи для адрес комп'ютерних мереж

<code>cidr</code>	IPv4 або IPv6 мережа
<code>inet</code>	IPv4 або IPv6 хост і мережа
<code>macaddr</code>	MAC адреса

#### Бітові рядки

<code>bit [ (n) ]</code>	Бітовий рядок фіксованої довжини
<code>bit varying [ (n) ]</code>	Бітовий рядок змінної довжини

#### Типи для пошуку тексту

<code>tsquery</code>	Запит на пошук тексту
<code>tsvector</code>	Список для пошуку тексту

#### UUID тип

<code>uuid</code>	Універсальний унікальний ідентифікатор
-------------------	--

#### XML типи

<code>xml</code>	данні XML
------------------	-----------

`Varchar (80)` задає тип даних, який може зберігати символальні рядки довжиною до 80 символів. `int` - це звичайний цілочисельний тип. `real` - це тип даних, що зберігає числа з плаваючою точкою одинарної точності. Тип `date` – поле введення дати. PostgreSQL підтримує такі типи SQL як `int`, `smallint`, `real`, `double precision`, `char (N)`, `varchar (N)`, `date`, `time`, `timestamp` та `interval`, а також і інші загальні типи і багатий набір геометричних типів. PostgreSQL можна налаштувати так, щоб він працював з довільним числом типів даних, визначених користувачем. Отже, імена типів не є синтаксичними ключовими словами, за виключенням тих випадків, де потрібні підтримка спеціально згідно стандартуSQL.

Приклад створення таблиці міста, яка містить назви міст і відповідні їм географічні координати:

```
CREATE TABLE cities (
    name varchar(80),
    location point
);
```

### 3.2 Внесення даних у таблицю

Для додавання записів в таблицю використовується команда `INSERT`:

```
INSERT INTO ім'я_таблиці [(спісок_імен_полів)] VALUES
(значення1, значення2, ..., значення n);
```

```
INSERT INTO weather VALUES
('Lviv', 46, 50, 0.25, '2014-11-27');
```

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					10

КР.П - 25.00.00.000 ПЗ

Важливо те, що всі типи даних, що використовуються в команді мають відповідні формати. Константи, які не є простими числовими значеннями зазвичай повинні бути укладені в одинарні лапки ('), як показано у прикладі. Тип date фактично може бути записаний по-різному, але ми будемо дотримуватися зрозумілого формату, який зображенний у прикладі.

### 3.3 Редагування таблиць

Під час **редагування таблиці** доводиться виконувати такі роботи:

- зміна даних, тобто редагування вмісту полів;
- маніпуляція записами, їх вилучення і вставка;
- перейменування та знищення таблиці;
- зміна структури таблиці, це вставка та вилучення полів, перейменування полів, додавання та вилучення обмежень тощо.

**Змінити дані** таблиці дозволяє команда:

```
UPDATE ім'я_таблиці SET ім'я_поля1=значення1, ім'я_поля2 =  
значення2, ... WHERE вираз
```

**Приклад.** Скласти команду для заміни в таблиці gazpr поля naz\_g на Україна тих записів, де kod\_g = 2:

```
UPDATE gazpr SET naz_g = 'Україна' WHERE kod_g = 2;
```

Якщо редагуються всі записи, то інструкція WHERE може бути опущена. Для нашого прикладу така команда матиме такий вигляд:

```
UPDATE gazpr SET naz_g = 'Україна' ;
```

**Знищення** всіх або вибраних за умовою записів таблиці забезпечує команда:

```
DELETE FROM ім'я_таблиці WHERE вираз
```

Приклад. Знищити ті записи таблиці gazpr, яких naz\_g = Україна.

```
DELETE FROM gazpr WHERE gazpr.naz_g = 'Україна' ;
```

**Вставку записів** забезпечує вже розглянута команда INSERT.

**Модифікація таблиці** командою ALTER TABLE. У більшості сучасних СУБД передбачена можливість модифікації таблиць командою ALTER TABLE. Її реалізація в PostgreSQL забезпечує такі види модифікацій: створення полів; призначення і скасування значень за умовчанням; перейменування таблиці; перейменування полів; додавання обмежень; зміна власника.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
11						

**Додавання полів.** Для створення нового поля в команду ALTER TABLE включається секція ADD COLUMN. Синтаксис команди ALTER TABLE з секцією ADD COLUMN:

**ALTER TABLE ім'я\_таблиці ADD [COLUMN] ім'я\_поля тип\_поля**

- таблиця - ім'я таблиці, в якій створюється нове поле;
- ім'я\_поля - ім'я нового поля;
- тип\_поля - тип нового поля.

Ключове слово COLUMN не є обов'язковим і включається в команду лише для наочності. Приклад включення в таблицю books нового поля publication для зберігання дати публікації:

**ALTER TABLE books ADD publication date;**

**Призначення і скасування значень за замовчуванням**

**ALTER TABLE таблиця ALTER [COLUMN] ім'я\_поля**

**{SET DEFAULT значення | DROP DEFAULT}**

Як і в попередньому розділі, ключове слово COLUMN є необов'язковим і включається в команду лише для наочності. Нижче наведено приклади встановлення та скасування послідовності значень за замовчуванням для поля id таблиці books.

**ALTER TABLE books ALTER COLUMN id SET DEFAULT nextval ('books.id');**

**ALTER TABLE books ALTER id DROP DEFAULT;**

**Перейменування таблиці** забезпечує команда:

**ALTER TABLE таблиця RENAME TO нове\_ім'я**

Таблицю можна перейменовувати скільки завгодно разів, це ніяк не відбивається на стані зберігаються в ній даних. Зрозуміло, що в деяких ситуаціях перейменування небажані, зокрема, якщо таблиця використовується зовнішнім додатком. Приклад зміни імені literature на books:

**ALTER TABLE literature RENAME TO books;**

**Перейменування полів.** PostgreSQL дозволяє змінювати імена полів без зміни даних, що зберігаються в таблиці. Втім, перейменування полів – справа ризикована, оскільки програми можуть містити посилання на імена полів. Якщо програма звертається до поля по імені, то перейменування може порушити її працездатність.

**ALTER TABLE таблиця RENAME [COLUMN] ім'я\_поля TO нове\_ім'я\_поля**

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

12

Як і в інших командах ALTER TABLE, ключове слово COLUMN є необов'язковим. Потрібно використати два ідентифікатора, розділеними ключовим словом TO, PostgreSQL може визначити, що команда перейменування відноситься до одного поля, а не до таблиці. Приклад:

```
ALTER TABLE daily RENAME COLUMN in_stock TO is_in_stock
```

**Додавання обмежень.** Після створення таблиці зберігаються деякі можливості додавання обмеження. У PostgreSQL команда ALTER TABLE з секцією ADD CONSTRAINT дозволяє визначати для полів існуючих таблиць тільки обмеження зовнішнього ключа. Команда створення нових обмежень має такий синтаксис:

```
ALTER TABLE таблиця ADD CONSTRAINT ім'я обмеження визначення
```

Синтаксис визначення залежить від типу обмеження. Нижче показано створення нового обмеження зовнішнього ключа для таблиці editions (пов'язаної з полем id таблиці books) і обмеження перевірки для поля type.

```
ALTER TABLE editions
```

```
ADD CONSTRAINT foreign_book FOREIGN KEY (bookid) REFERENCES books (id);
```

```
ALTER TABLE editions
```

```
ADD CONSTRAINT hard_or_paper_back CHECK (type = 'p1 OR type = ' h '');
```

Установка обмеження зовнішнього ключа призводить до того, що будь-яке значення book\_id у таблиці editions також має існувати і в таблиці books. Крім того, внаслідок встановленого обмеження перевірки полі type в таблиці editions може містити тільки значення p або b. При створенні унікального індекса командою CREATE INDEX також неявно встановлюється і обмеження унікальності.

В існуючу таблицю можна додати тільки обмеження таблиць. Це робить така команда:

```
ALTER TABLE ім'я_таблиці
```

```
ADD [ CONSTRAINT ім'я_та_визначення_обмеження ]
```

Команда ALTER додає обмеження таблиці, але не поля. Нижче встановлюється нове обмеження FOREIGN KEY для поля kod\_g таблиці gazpr, яке зв'язується з полем kod\_g таблиці oblik. Обмеження FOREIGN KEY гарантує, що внаслідок вставки або оновлення даних у поле kod\_g таблиці oblik не з'являться значення, відсутні в полі kod\_g таблиці gazpr, тому у прикладі система видала відповідне повідомлення про помилку, тобто про те, що в таблиці gazpr не існує газопроводу з кодом 7.

```
ALTER TABLE gazpr ADD CONSTRAINT bbb PRIMARY KEY (kod_g);
```

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						13

```

ALTER TABLE oblik ADD CONSTRAINT aaa
    FOREIGN KEY (kod_g) REFERENCES gazpr(kod_g);
    INSERT INTO OBLIK VALUES(4,4,'21.12.2000',4);
    INSERT INTO OBLIK VALUES(7,4,'21.12.2000',4);

```

*ERROR: insert or update on table "oblik" violates foreign key constraint "aaa"* **DETAIL:** Key (kod\_g)=(7) is not present in table "gazpr". SQL state: 23503      Обмеження унікальності також неявно встановлюється при створенні унікального індекса командою CREATE INDEX.

**Видалення обмежень.** Деякі версії PostgreSQL не підтримують пряме видалення обмежень із таблиці. Добитися потрібного результату можна лише одним способом: створити копію таблиці, що практично повністю повторює оригінал, який не містить тих обмежень, які видаляються. Дані копіюються з початкової таблиці в нову, після чого таблиці перейменовуються командою ALTER TABLE і копія замінює оригінал.

**Зміна власника.** За замовчуванням творець таблиці автоматично стає її власником. Власник володіє всіма правами, пов'язаними з таблицею, в тому числі правами передання і відкликання прав командами GRANT і REVOKE (будуть розглянені пізніше у відповідному розділі). Зміна власника проводиться командою ALTER TABLE з секцією OWNER. Команда має такий синтаксис:

```
ALTER TABLE ім'я_таблиці OWNER TO новий_власник
```

Змінювати власника таблиці може або поточний власник, або суперкористувач. Приклад зміни власника таблиці employee, де новим власником стає користувач corwin:

```
ALTER TABLE employees OWNER TO corwin;
```

**Реструктуризація таблиць командами CREATE TABLE і INSERT INTO.** Якщо таблиця, створена командою CREATE TABLE AS, не влаштовує (наприклад, якщо в таблиці необхідно встановити обмеження полів), то одну команду CREATE TABLE AS можна замінити двома командами SQL. Спочатку команда CREATE TABLE створює нову таблицю, а потім команда INSERT INTO з запрограмованим SELECT заповнює її даними. Приклад реструктуризації таблиці командами CREATE TABLE і INSERT INTO:

```

CREATE TABLE newbooks (
    id integer UNIQUE,
    title text NOT NULL,
    authorid integer,
    subjected integer CONSTRAINT books_id_pkey PRIMARY KEY (id));

```

*NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index, books\_id\_pkey1  
for table 'new\_books'*

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						14

```

INSERT INTO new_books
SELECT id, title, author_id, subjected FROM books;

ALTER TABLE books RENAME TO old_books;
ALTER TABLE new_books RENAME TO books;

```

### Видалення таблиць забезпечує команда: **DROP TABLE** таблиця

Використання команди **DROP TABLE** вимагає обережності, оскільки видалення таблиці спричиняє знищення всіх збережених у ній даних. При знищенні таблиці з неявно створеним індексом знищуються і всі пов'язані з нею індекси.

**Системні поля.** У PostgreSQL всі таблиці містять системні поля, які залишаються невидимими для користувача і не виводяться при вибірці (якщо службова інформації не запитується спеціально). У системних полях зберігаються метадані, які описують вміст записів, вони такі:

Поле	Опис
oid	4-байтовий унікальний ідентифікатор запису
tableoid	Ідентифікатор таблиці. Ім'я таблиці зв'язується з ідентифікатором у системній таблиці pg_class
xmin	Ідентифікатор транзакції вставки для кортежу
cmin	Ідентифікатор команди, асоційованої з транзакцією вставки для кортежу
xmax	Ідентифікатор транзакції видалення для кортежу. Для видимих (не видалених) кортежів дорівнює нулю
cmax	Ідентифікатор команди, асоційованої з транзакцією видалення для кортежу. За аналогією з xmax дорівнює нулю для видимих кортежів
ctid	Ідентифікатор, що описує фізичне місце знаходження кортежу в БД. Поле ctid містить пару чисел: номер блоку та індекс кортежу в блоці

**Ідентифікатори записів** дозволяють розрізнати два записи з одинаковими значеннями полів. Для цього в PostgreSQL передбачений ідентифікатор OID, унікальний в межах таблиці.

Приклад: **SELECT \*, oid FROM my\_list;**

Знищенння таблиці виконує команда:

**DROP TABLE tablename;**

### 3.4 Обмеження

PostgreSQL має декілька варіантів обмеження даних (constraint), які впливають на операції вставки і оновлення. Розглянемо один із них, який полягає в установці *обмежень* для таблиць і полів. *Обмеженням* є особливий атрибут таблиці, який встановлює критерії допустимості для вмісту її полів. Дотримання цих правил допомагає запобігти заповненню бази помилковими або невідповідними даними. Обмеження задаються в секції CONSTRAINT при створенні таблиці командою CREATE TABLE. Обмеження полів завжди стосуються лише

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						15

одного поля, тоді як обмеження таблиць можуть встановлюватися як для одного, так і для декількох полів. У команді CREATE TABLE обмеження полів задаються відразу ж після визначення поля, тоді як обмеження таблиці встановлюється в спеціальному блоці, виділеному комами від усіх визначень полів.

Опис обмеження поля виглядає так:

```
[ CONSTRAINT обмеження ]
{ NOT NULL | UNIQUE | PRIMARY KEY | DEFAULT значення CHECK (умова) |
REFERENCES таблиця [ ( поле ) ]
[ MATCH FULL | MATCH PARTIAL ]
[ ON DELETE операція]
[ ON UPDATE операція]
[ DEFERRABLE | NOT DEFERRABLE ]
[ INITIALLY DEFERRED | INITIALLY IMMEDIATE ] }
```

Визначення обмеження слідує в команді CREATE TABLE відразу ж за типом обмежуваного поля і передує комі, яка відокремлює його від наступного поля. Обмеження можуть встановлюватися для будь-якої кількості полів, а ключове слово CONSTRAINT і ідентифікатор *обмеження* не обов'язкові.

Існує шість типів обмежень полів, які задаються за допомогою спеціальних ключових слів. Деякі з них побічно встановлюються при створені обмежень іншого типу. Типи обмежень полів перераховані нижче, вони такі:

- NOT NULL. Поле не може містити псевдозначення NULL. Обмеження NOT NULL еквівалентно обмеженню CHECK (*поле* NOT NULL);
- UNIQUE. Поле не може містити значення, які повторюються. Слід враховувати, що обмеження UNIQUE допускає багатократне входження псевдозначень NULL, оскільки формально NULL не збігається ні з яким іншим значенням;
- PRIMARY KEY. Автоматично встановлює обмеження UNIQUE і NOT NULL, а для заданого поля створюється індекс. У таблиці може встановлюватися тільки одне обмеження первинного ключа;
- DEFAULT **значення**. Пропущені значення поля замінюються заданою величиною. Тип значення за замовчуванням повинно відповідати типу поля;
- CHECK **умова**. Команда INSERT або UPDATE завершується успішно лише при виконанні заданої умови (виразу, що повертає логічний результат). При установці обмеження поля в секції CHECK може використовуватися тільки поле, для якого встановлюється обмеження;
- REFERENCES. Це обмеження складається з таких секцій:
  - 1 – REFERENCES **таблиця** [(*поле*)]. Вхідні значення обмежуваного поля порівнюються із значеннями іншого поля в заданій таблиці. Якщо збіги відсутні, то команда INSERT

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						16

або UPDATE завершується невдачею. Якщо параметр *поле* не вказаний, то перевірка виконується за першним ключем. Обмеження REFERENCES подібне до обмеження таблиці FOREIGN KEY, яке описане в наступному пункті цього підрозділу. Дійсно, між цими обмеженнями є багато спільного;

2–MATCH FULL | MATCH PARTIAL. Секція MATCH указує, чи дозволяється змішувати значення NULL і звичайні значення при вставці в таблицю, у якої зовнішній ключ посилається на *декілька* полів. Таким чином, на практиці секція MATCH приносить користь лише в обмеженнях таблиць, хоча формально вона може використовуватися і при обмеженні полів. Конструкція MATCH FULL забороняє вставку даних, у яких частина полів зовнішнього ключа містить псевдозначення NULL (крім випадку, коли NULL міститься у всіх полях). У PostgreSQL 7.1.x конструкція MATCH PARTIAL не підтримується. Якщо секція MATCH відсутня, то вважається, що поля з псевдозначеннями NULL задовільняють обмеження. Також буде доречно нагадати, що обмеження полів стосується лише *одного* поля, тому секція MATCH використовується лише в обмеженнях таблиць;

3–ON DELETE *операція*. При виконанні команди DELETE для заданої таблиці з обмежуваним полем виконується одна з таких операцій:

NO ACTION (якщо видалення спричиняє порушення цілісності посилань, то відбувається помилка; використовується за замовчуванням, якщо операція не вказана), RESTRICT (аналогічно NO ACTION),

CASCADE (видалення всіх записів, що містять посилання на запис, який видаляється), SET NULL (поля, що містять посилання на запис, який видаляється, замінюються псевдозначеннями NULL),

SET DEFAULT (полям, що містять посилання на запис, який видаляється, привласнюється значення за замовчуванням);

4–ON UPDATE *операція*. При виконанні команди UPDATE для заданої таблиці виконується одна з вищеописаних операцій. За замовчуванням використовується значення NO ACTION. Якщо вибрана операція CASCADE, всі записи, які містять посилання на оновлюваний запис, оновлюються новим значенням (замість видалення, як у випадку з ON DELETE CASCADE);

5–DEFERRABLE | NOT DEFERRABLE. Значення DEFERRABLE дозволяє відкласти виконання обмеження до кінця транзакції (замість негайного виконання після завершення команди). Значення NOT DEFERRABLE означає, що обмеження завжди перевіряється відразу ж після завершення чергової команди. В цьому випадку

Змн.	Арк.	№ докум.	Підпис	Дата

користувач не може відкласти перевірку обмеження до кінця транзакції. За замовчуванням вибирається саме цей варіант;

6–INITIALLY DEFERRED | INITIALLY IMMEDIATE. Секція INITIALLY задається для обмежень, визначених з ключовим словом DEFERRED. Значення INITIALLY DEFERRED відкладає перевірку обмеження до кінця транзакції, а при установці значення INITIALLY IMMEDIATE перевірка проводиться післяожної команди. За відсутності секції INITIALLY за замовчуванням використовується значення INITIALLY IMMEDIATE. У прикладі нижче показано створення таблиці gazpr з декількома обмеженнями.

```
CREATE TABLE gazpr
(kod_g integer PRIMARY KEY CHECK (kod_g > 0),
naz_g CHARACTER(80) NOT NULL,
kilnyt INTEGER CHECK (kilnyt > 1),
diam INTEGER);
```

*NOTICE: CREATE TABLE/PRIMARY KEY will create implicit index 'gazpr\_pkey' for table 'gazpr'*

Тут створюється поле kod\_g типу integer, для якого встановлюються обмеження PRIMARY KEY і CHECK. Обмеження PRIMARY KEY також має на увазі обмеження NOT NULL і UNIQUE і зумовлює автоматичне створення індекса gazpr\_pkey для обмежуваного поля. Обмеження CHECK гарантує, що значення поля kod\_g завжди буде додатнє, а поля kilnyt – більше за одиницю. Це означає, що будь-які спроби вставки або оновлення в таблиці gazpr записів, поле kod\_g яких менше або дорівнює 0, завершуються невдачою.

Таблиця gazpr також містить текстове поле naz\_g, для якого встановлено обмеження NOT NULL. Воно забороняє мати в таблиці записи, поле naz\_g яких містить NULL. Інакше кажучи, це поле обов’язкове для заповнення.

**Обмеження таблиць** подібні до обмежень полів, але на відміну від них, можуть стосуватися відразу декількох полів таблиці.

Нижче створюються дві таблиці: gazpr і oblik з трьома обмеженнями:

```
CREATE TABLE gazpr
(kod_g INTEGER, naz_g CHARACTER(80), kilnyt INTEGER, diam INTEGER,
CONSTRAINT aaa PRIMARY KEY (kod_g), CONSTRAINT bbb CHECK (kilnyt IS
NOT NULL AND diam IS NOT NULL);
```

*NOTICE: CREATE TABLE/PRIMARY KEY will create implicit index 'pkey' for table 'gazpr'*

```
CREATE TABLE oblik (kod_g INTEGER, kod_p INTEGER, data DATE, kilk
REAL, CONSTRAINT ccc FOREIGN KEY (kod_g) REFERENCES gazpr(kod_g)
ON DELETE CASCADE ON UPDATE CASCADE);
```

*NOTICE: CREATE TABLE will create implicit trigger(s) for FOREIGN KEY check(s)*

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					18

Перше обмеження, ааа належить до типу PRIMARY KEY і встановлюється для поля kod\_g. Воно практично не відрізняється від обмеження PRIMARY KEY для поля, оскільки список у даному прикладі складається всього з одного поля.

Обмеження bbb гарантує, що поля kilnyt і diam не буде містити псевдозначення NULL.

Нарешті, обмеження ccc таблиці oblik за допомогою конструкцій FOREIGN KEY і REFERENCES гарантує, що значення поля kod\_g таблиці oblik зустрічаються і в полі kod\_g таблиці gazpr. Більш того, оскільки в секціях ON DELETE і ON ACTION зустрічається ключове слово CASCADE, будь-які модифікації поля kod\_g в таблиці gazpr спричинять каскадні зміни записів у таблиці oblik, а при видаленні записів з таблиці gazpr будуть видалені відповідні записи таблиці oblik.

Деякі версії PostgreSQL забезпечують лише додавання обмежень CHECK і FOREIGN KEY. Вибір режиму перевірки обмежень забезпечує команда:

```
SET CONSTRAINTS { ALL | режим [....] } { DEFERRED | IMMEDIATE },  
де:
```

- ALL означає, що вказаний режим повинен стосуватися всіх обмежень;
- *режим* – ім'я обмеження, для якого встановлюється режим перевірки;
- DEFERRED – перевірка обмежень (або конкретного обмеження) вікладається до моменту фіксації транзакції, тобто до виконання команди COMMIT;
- IMMEDIATE – всі або конкретне обмеження перевіряються наприкінці *кожної* команди.

Команда SET CONSTRAINTS задає режим перевірки для всіх обмежень або одного обмеження в поточному транзакційному блоці. Існує два режими перевірки: негайна (IMMEDIATE) і відкладена (DEFERRED) перевірка обмежень. У режимі IMMEDIATE всі обмеження перевіряються після виконання *кожної* команди транзакції, а в режимі DEFERRED обмеження перевіряються лише після виконання команди COMMIT.

Зауважимо, що деякі версії PostgreSQL забезпечують зміну режиму перевірки тільки для обмеження FOREIGN KEY. Команда SET CONSTRAINTS не разповсюджується на обмеження CHECK і UNIQUE.

Нижчеприведена команда вибирає режим негайної перевірки всіх обмежень.

```
SET CONSTRAINTS ALL IMMEDIATE;
```

### 3.5 Послідовності

PostgreSQL є об'єктно-реляційною СУБД, що дозволило включити в неї ряд нестандартних розширень SQL. Частина цих розширень пов'язана з автоматизацією часто вживаних операцій з базами даних, це, зокрема, послідовності і тригери.

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					19

КР.П - 25.00.00.000 ПЗ

**Послідовність** (sequence) являє собою послідовність чисел. Значення послідовності мають тип integer, тому її значення повинні лежати в межах від 2 147 483 647 до -2 147 483 647. Як правило, її використовують у обмеженні DEFAULT для надання унікальних значень полям таблиць. Ці значення автоматично змінюються шляхом виклику функції nextval().

Послідовність створюється командою:

**CREATE SEQUENCE** послідовність

[ **INCREMENT** приріст ] [ **MINVALUE** мінімум ] [ **MAXVALUE** максимум ]  
[ **START** початок ] [ **CACHE** кеш ] [ **CYCLE** ]

де:

- *послідовність* – ім’я послідовності, це єдиний обов’язковий параметр;
- **INCREMENT** *приріст* – нарощення поточного значення послідовності, це число може бути від’ємним або додатнім. За замовчуванням приріст дорівнює 1.
- **MINVALUE** *мінімум* – мінімально допустиме значення. Спроба зменшити поточне значення нижче за заданий мінімум спричинить помилку або циклічний перехід до максимального значення (якщо послідовність створювалася з ключовим словом CYCLE). За замовчуванням мінімальне значення дорівнює 1;
- **MAXVALUE** *максимум* – максимально допустиме значення послідовності. Спроба перевищити заданий максимум спричинить помилку або циклічний перехід до мінімального значення (при CYCLE). За замовчуванням максимальне значення дорівнює 2 147 483 647;
- **START** *початок* – початкове значення послідовності, це будь-яке ціле число в інтервалі між мінімальним і максимальним значеннями;
- **CACHE** *кеш* – забезпечує можливість попереднього обчислення і зберігання значень послідовності в оперативній пам’яті. Кешування прискорює доступ до тих послідовностей, які часто використовуються;
- **CYCLE** – повторне, циклічне використання послідовності. Досягши нижнього або верхнього порога, послідовність продовжує генерувати нові значення, тоді вона переходить до мінімального значення при зростанні послідовності або до максимального – при убуванні. Оскільки в цих випадках значення послідовності повторюються, то вони не будуть унікальними.

У нижчеприведеному прикладі створюється послідовність з ім’ям ship, яка починається із значення 0 і збільшується на 1 до тих пір, поки не досягне максимального значення. Ключове слово CYCLE не вказане, тому ця послідовність набуває лише унікальних значень.

**CREATE SEQUENCE** ship **MINVALUE** 0 ;

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

20

До послідовності також можна звернутися командою SELECT, як до таблиці, хоча така можливість використовується відносно рідко, наприклад, під час випробування або перевірки послідовності. При складанні запиту до послідовності в списку вибірки вказуються її атрибути, перелічені в таблиці 3.1.

Таблиця 3.1. Атрибути послідовності

Атрибут	Тип	Примітка
sequence_name	name	Ім'я послідовності
last_value	integer	Поточне значення
increment_by	integer	Нарощення
max_value	integer	Максимальне значення
min_value	integer	Мінімальне значення
cache_value	integer	Об'єм кеша, байт
log_cnt	integer	
is_cycled	"char"	
is_called	"char"	

Нижче показано запит до послідовності ship.

```
SELECT last_value, increment_by FROM ship;
```

Як правило, всі операції з послідовностями виконуються за допомогою таких трьох спеціальних функцій PostgreSQL

- nextval(‘послідовність’) – нарощує поточне і повертає нове значення;
- curval(‘послідовність’) – повертає поточне значення;
- setval(‘послідовність’, n) – змінює поточне значення на число n.

Нижче в прикладі виводиться пара чергових значень послідовності ship.

Послідовності найчастіше використовуються для задавання значень за замовчуванням у полях таблиць. Нехай таблицю pidpr створено командою:

```
CREATE TABLE pidpr(kod_p integer PRIMARY KEY DEFAULT
nextval('ship') , naz_p character(50) , misto character(30));
```

Тут поле kod\_p має обмеження DEFAULT, функція nextval('ship') якого буде надавати кожного разу нарощені на одиницю значення за замовчуванням, вироблені послідовністю ship. Це поле має також обмеження PRIMARY KEY, яке, нагадаємо, означає ще й обмеження UNIQUE, тому в нього неможливо буде помилково занести неунікальні значення.

Тепер команда вставки даних у таблицю pidpr може виглядати, наприклад, так:

```
INSERT INTO pidpr (naz_p, misto) VALUES
('Уренгой – Помари – Ужгород', 'Полтава');
```

Під час виконання команди код pidpr буде занесений у таблицю автоматично.

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					21

Змінити поточне значення послідовності дозволяє функція `setval()`, нижче показано, що значення послідовності `ship` змінюється на 1010, а далі нарощується на 1, тому дорівнює 1011:

```
SELECT setval('ship', 1010);
```

**Видалення послідовності** забезпечує команда SQL `DROP SEQUENCE`, вона видаляє одну або декілька послідовностей одночасно. Ця команда має такий вигляд:

```
DROP SEQUENCE перелік_послідовностей
```

Приклад видалення послідовності `ship`: `DROP SEQUENCE ship;`

Перш ніж знищувати послідовність, слід переконатися в тому, що вона не використовується іншою таблицею, функцією або іншим об'єктом бази даних. Якщо забути про цю перевірку, то можна порушити роботу інших операцій, залежних від даної послідовності. Вивести імена всіх таблиць, в яких використовується задана послідовність можна за допомогою такого запиту:

```
SELECT p.relname, a.adsrc FROM pg_class p JOIN pg_attrdef a ON  
(p.relfilenode = a.adrelid) WHERE a.adsrc ~ '"sequence_name"';
```

Приклад пошуку імен таблиць, у яких використовується послідовність `ship`.

```
SELECT p.relname, a.adsrc FROM pg_class p JOIN pg_attrdef a ON  
(p.relfilenode = a.adrelid) WHERE a.adsrc ~ '"ship"';
```

### 3.6 Запити

Центральне місце в SQL займає команда `SELECT`, призначена для побудови запитів та вибірки даних з таблиць. Дані, повернені в результаті запиту, називаються підсумковим набором. Дані підсумкового набору не зберігаються на диску в постійній формі. Підсумковий набір є лише тимчасовим поданням даних, отриманих в результаті запиту.

У найпростішому випадку команда `SELECT` вибирає з заданої таблиці всі дані. Повна вибірка даних проводиться командою `SELECT * FROM імя_таблиці`.

**Запит** являє собою команду, написану мовою SQL, яка вибирає дані з таблиць. Якщо запит виготовляється візуальним способом, то інтерпретатор складає його скрипт у вигляді тексту, написаний теж мовою SQL. Скрипт вигідний тим, що його можна редагувати, він також застосовується на стадії вивчення мови SQL. На відміну від інших мов програмування, запит вказує серверу, які дані потрібно знайти, але не повідомляє, як це зробити. Слід особливо наголосити на тому, що запит – команда, але не вибірка даних.

Дані, повернені запитом, називаються **підсумковим набором**. Як і таблиця, він складається із записів та полів, тобто виглядає так, як таблиця. Вибірка не дублює таблицю, але містить її задану частину або всю інформацію, вона формується заново при кожному

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						22

відкритті (запуску) запиту. Підсумковий набір не зберігається на диску, це тимчасові дані. Він може містити поля як із однієї, так з декількох таблиць.

Загальний варіант команди SELECT:

```
SELECT список імен полів запиту FROM ім'я джерела
INNER JOIN ім'я зв'язаного джерела ON ім'я поля = ім'я зв'язаного
поля
WHERE ім'я поля, оператор, значення
GROUP BY список імен полів, у яких відбувається групування
HAVING умова відбору записів
ORDER BY ім'я поля, за даними якого записи запиту будуть
посортовані;
```

Тут службові слова SELECT і FROM – обов'язкові.

WHERE дозволяє обмежити набір записів. Якщо використовуються всі поля джерела, то замість їх переліку у списку імен полів запиту вживається символ зірочки.

Секція GROUP BY дозволяє групувати записи за заданим критерієм (наприклад, підсумувати кількість транспортуваного газу для кожного газопроводу).

Приклад простого запиту: видати всі поля всіх записів таблиці gazpr, де діаметр однієї нитки (поле diam) менший за 1800:

```
SELECT * FROM gazpr WHERE diam < 1800;
```

Як бачимо, запит може містити **вирази**. До складу виразу можуть входити: константи; оператори; імена полів і таблиць; функції; фігурні, квадратні і круглі дужки.

**Оператори** зазвичай містять від одного до чотирьох символів. Слід зауважити, що деякі з них існують тільки у версії SQL для PostgreSQL і можуть не бути в інших СУБД. Об'єднання двох рядкових констант у один виконується оператором **конкатенації** (||), наприклад, так:

```
SELECT 'Pa' || 'кета' AS example;
```

Крім, власне демонстрації злиття двох рядків, у цьому прикладі показано застосування **альтернативного імені** example, яке тут дозволяє підвищити наглядність результату.

**Строкові оператори** служать для порівняння двох рядків (або чисел), вони такі:

Оператор	Опис
=	Повертає True, якщо перший рядок точно збігається з другим
!=	Повертає True, якщо перший рядок не збігається з другим
<>	Ідентичний оператору !=
<	Повертає True, якщо перший рядок передує другому
<=	Повертає True, якщо перший рядок передує другому або їх значення співпадають
>	Повертає True, якщо другий рядок передує першому
>=	Повертає True, якщо другий рядок передує першому або їх значення співпадають

**Оператори регулярних виразів** розширяють можливості строкових операторів, вони такі:

~ – перевіряє (відає true, якщо так) чи в рядку існує збіг для регулярного виразу;

Змн.	Арк.	№ докум.	Підпис	Дата

- $!~$  – перевіряє чи в рядку немає збігу для регулярного виразу;
- $\sim*$  – перевіряє чи в рядку існує збіг без урахування регістру символів;
- $!~*$  – перевіряє чи в рядку немає збігу без урахування регістру символів.

Регулярний вираз може містити такі метасимволи (спеціальні символи):

- $^$  – з початку рядка;
- $$$  – наприкінці рядка;
- $.$  – один символ;
- $[ ]$  – будь-який із символів, перелічених у квадратних дужках;
- $[^]$  – будь-який символ, крім перелічених у квадратних дужках;
- $[-]$  – будь-який символ з інтервалу, заданого в квадратних дужках;
- $[^ -]$  – будь-який символ, крім символів з інтервалу, заданого в квадратних дужках;
- $?$  – один примірник попереднього символа або підвиразу (якщо, звичайно, він є) ;
- $*$  – довільна кількість примірників попереднього символа або підвиразу;
- $+$  – один і більше примірників попереднього символа або підвиразу;
- $|$  – лівий або правий підвираз;
- $( )$  – групування підвиразів з явним визначенням пріоритету операцій;

Нижче наведено декілька прикладів застосування регулярних виразів для видачі:

- всіх значень поля name таблиці authors, які починаються з літер A або T:

```
SELECT name FROM authors WHERE name ~ '^A|^T';
```

- всіх значень поля title тих записів таблиці books, де зустрічається слово “Кобзар”:

```
SELECT title FROM books WHERE title ~* 'Кобзар' ;
```

- всіх значень, які починаються (^) на “Кобзар” без врахування регистра (\*), а далі через довільне число символів(.) містять слово “малий” або (|) “повний”, написане лише малими буквами (перед ними немає зірочки):

```
SELECT title FROM books WHERE title ~* '^ Кобзар.( малий | повний)' ;
```

- всіх значень, які починаються (^) на “К” без врахування регистра (\*), а далі через довільне число символів(.) зустрічається якась буква з переліку ([] “о” та “м”, наступним символом після яких зразу ж стоїть буква “к”, або (|) видати всі ті значення, які закінчуються (\$) на “ий” або (|) на “не”:

```
SELECT title FROM books WHERE title ~* ' (^K.*[о,м]к) | (ий$ | не$)' ;
```

**Математичні оператори** використовуються в числових виразах, вони такі:

$a + b$	Підсумування чисел a і b
$a - b$	Віднімання числа b від a
$a * b$	Множення числа a на b
$a / b$	Ділення числа a на b
$a \% b$	Залишок від ділення a на b

Змн.	Арк.	№ докум.	Підпис	Дата

$a ^ b$	Піднесення а до степеня b
$\sqrt{a}$	Квадратний корінь з а
$\sqrt[3]{a}$	Кубічний корінь з а
$a!$	Факторіал а
$!a$	Факторіал а (відрізняється від постфіксного оператора тільки розташуванням аргумента а)
$@a$	Модуль (абсолютне значення) а

Приклад застосування операції множення: видати коди газопроводів та кількість транспортуваного газу за весь період обліку, збільшенну в 1.6 разів (так начисляється премія персоналу, який обслуговує газопровід) і названу premia, взяті з таблиці oblik:

```
SELECT kod_g, (1.6 * kilk)::numeric(8, 2) AS premia FROM oblik;
```

**Математичні оператори порівняння** такі ж, як вище розглянуті строкові.

Ключове слово **BETWEEN** (також іноді називається оператором) дозволяє перевірити, чи входить значення в деякий інтервал. Наприклад, нище наведена команда SELECT видає книги, ціна яких знаходиться в інтервалі між 10 та 17:

```
SELECT naz_kniga FROM kniga WHERE cina BETWEEN 10 AND 17;
```

**Двійкові оператори** виконують порозрядні операції з бітовими послідовностями або цілими числами, вони аналогічні до мови С. Приклад зсуву числа 8 (двійкове 1000) на два розряди вправо та перетворення одержаної бітової послідовності в ціличисловий тип функцією bittoint4():

```
SELECT bittoint4(b'1000' >> 2);
```

**Пріоритет операторів SQL** (у порядку спадання):

Оператор	Опис
::	Явне перетворення типу
[ ]	Індексація елемента масиву
.	Крапка , роздільник імен таблиці і поля
-	Унарний мінус
^	Піднесення до степеня
* / %	Множення, ділення і залишок
+ -	Додавання і віднімання
IS значення	Перевірка наявності заданого значення
IS NULL	Перевірка наявності значення NULL
IS NOT NULL	Перевірка розбіжності з NULL
IN перелік	Задання набору значень
BETWEEN	Перевірка приналежності до інтервалу
LIKE ILIKE	Вибір за зразком
<> <= > =	Оператори порівняння
AND OR NOT	Логічні оператори

### 3.7 Об'єднання таблиць

Запити можуть звертатися не тільки до однієї, а одночасно до декількох таблиць або до в один і той же час до тієї ж таблиці але з іншим способом обробки довільних записів. Запит,

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					25

який звертається до кількох записів однієї таблиці або до декількох таблиць одночасно, називається *join* запитом (об'єднанням). Як приклад, скажімо ви хочете подивитися всі записи про погоду разом з тими містами, до яких вони належать. Щоб це зробити, потрібно порівняти поле city кожного запису про погоду в таблиці погоди weather з ім'ям поля всіх записів в таблиці міст cities, і вибрати пари записів, де ці значення збігаються.

Це тільки концептуальна модель. Фактично, об'єднання може бути виконано більш ефективно, але це буде не видно користувачеві.

Таким чином, потрібна нам дію буде виконано наступним запитом:

```
SELECT * FROM weather, cities WHERE city = name;
```

city	temp_lo	temp_hi	prcp	date	name	location
San Francisco	46	50	0.25	1994-11-27	San Francisco	(-194,53)
San Francisco	43	57	0	1994-11-29	San Francisco	(-194,53)

Після виконання якого ми отримаємо два рядки:

- Тут немає запису для міста Hayward, тому що для цього міста немає відповідного запису в таблиці cities, і таким чином, об'єднання ігнорує незбіжні записи в таблиці weather.
- В отриманому результаті є два поля, що містять ім'я міста. Це правильно, тому що відбувається злиття списку полів таблиць weather і cities. На практиці це небажано, і тому треба явно вказати ті поля, які потрібно, замість використання :

```
SELECT city, temp_lo, temp_hi, prcp, date, location
      FROM weather, cities WHERE city = name;
```

Оскільки всі поля мають різні імена, аналізатор автоматично знаходить яке ім'я відповідає якій таблиці, але хорошим стилем у запитах об'єднання є вказівка повного імені поля:

```
SELECT weather.city, weather.temp_lo, weather.temp_hi,
       weather.prcp, weather.date, cities.location
      FROM weather, cities
     WHERE cities.name = weather.city;
```

Запити об'єднання такого виду, можуть бути також записані в альтернативній формі, яку значно доцільніше використовувати, ніж попередню:

```
SELECT * FROM weather INNER JOIN cities ON
                  (weather.city = cities.name);
```

Тепер розглянемо як можна отримати записи для міста Hayward. Для цього потрібен такий запит, який би виконав сканування таблиці weather і для кожного запису в ній, знайшов відповідний запис у таблиці cities. Якщо відповідний запис не знайдено, ми хочемо підставити деякі "порожні значення" на місце полів таблиці cities. Такий вид запитів називається *outer join* (зовнішнє об'єднання). Ось як буде виглядати запит:

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						26

```
SELECT * FROM weather LEFT OUTER JOIN cities
    ON (weather.city = cities.name);
```

Даний запит називається **left outer join**, тому що для таблиці згадується в лівій частині оператора об'єднання join в отриманому результаті буде видано всі записи принаймні один раз, у той час як для таблиці праворуч в результат потраплять тільки ті записи, які відповідають записам у таблиці ліворуч. Коли буде проводиться висновок записи з таблиці ліворуч, для якої немає відповідного запису в таблиці праворуч, в поля, відповідні таблиці праворуч будуть підставлені порожні (null) значення.

Також можна об'єднувати таблицю само з собою. Такий вид запитів називається **self join**. Припустимо, що ми хочемо знайти всі записи про погоду, які знаходяться в певному діапазоні температур. Для цього нам необхідно порівняти значення полів temp\_lo і temp\_hi для кожного запису про погоду weather з значеннями полів temp\_lo і temp\_hi для всіх інших записів про погоду weather. Це можна зробити за допомогою наступного запиту:

```
SELECT W1.city, W1.temp_lo AS low, W1.temp_hi AS high,
       W2.city, W2.temp_lo AS low, W2.temp_hi AS high
  FROM weather W1, weather W2
 WHERE W1.temp_lo < W2.temp_lo
   AND W1.temp_hi > W2.temp_hi;
```

У цьому запиті переназвано таблицю weather іменами W1 і W2 щоб розрізняти ліву і праву частини оператора об'єднання join. Також можна використовувати такі види псевдонімів в інших запитах, щоб зробити запис запиту коротшим, тобто:

```
SELECT * FROM weather w, cities c WHERE w.city = c.name;
```

Такий стиль абревіатур зустрічається часто.

### 3.8 Агрегатні функції

PostgreSQL, як і багато інших реляційні СУБД, підтримує агрегатні функції. Агрегатна функція проводить обчислення над одиничним результатом від безлічі записів. Наприклад, є агрегати для обчислення count(кількості), sum (суми), avg (середнього арифметичного), max (максимального значення) і min (мінімального значення) списку записів.

**Агрегатні функції** застосовуються в запитах для групування даних, вони такі:

avg(вираз) – середнє арифметичне значення;

count(вираз) – кількість записів зі значенням відмінним від NULL;

max(вираз) – максимальне значення;

min(вираз) – мінімальне значення;

stddev(вираз) – середньоквадратичне відхилення;

Змн.	Арк.	№ докум.	Підпис	Дата

`sum(вираз)` – сума;

`variance(вираз)` – дисперсія.

Як приклад, можна знайти найбільш високу низьку температуру, створивши запит:

```
SELECT max(temp_lo) FROM weather;
```

Якщо необхідно знати, в якому місті (або містах) це відбувалося, можна спробувати створити такий запит:

```
SELECT city, max(temp_lo) FROM weather GROUP BY city;
```

В даному випадку підзапит є незалежним обчисленням, яке обчислює свій власний агрегат окремо від того, який працює в зовнішньому запиті. Агрегати є дуже корисними в комбінаціях з літералами GROUP BY. Наприклад, ми можемо отримати максимальну низьку температуру, зазначену в кожному місті за допомогою запиту:

```
SELECT city, max(temp_lo) FROM weather GROUP BY city;
```

який надасть нам по одному запису на місто. Кожен результат агрегату підраховується виходячи із записів таблиці, які відповідають певному місту. Можна фільтрувати згруповано записи, використовуючи літерал HAVING:

```
SELECT city, max(temp_lo) FROM weather
GROUP BY city HAVING max(temp_lo) < 40;
```

### 3.9 Курсори SQL

Курсор мовою SQL являє собою вказівник на підсумковий набір даних, виданих командою SELECT. Курсор виконується тільки в транзакційному блоці. Його використання надає такі переваги:

- економія оперативної пам'яті за рахунок зберігання тільки потрібних даних (вибірки). Це особливо вигідно під час використання громіздких таблиць;
- зменшення навантаження на зовнішні носії пам'яті при багаторазовому читанні вибірки, включаючи читання одних і тих же даних. Це дозволяє зменшити кількість звертань до зовнішньої пам'яті при багаторазовому використанні одних і тих же даних. Зменшується навантаження й на мережу в багатокористувальських системах;
- збільшення швидкості звертання до даних за рахунок уникнення блокування.

**Оголошення курсора** відбувається тільки в складі транзакційного блока, який починається з команди BEGIN. В SQL команда оголошення курсора одночасно виконує його відкриття, вона має такий вигляд:

```
DECLARE курсор [ BINARY ] [ INSENSITIVE ] [ SCROLL ]
CURSOR FOR запит
```

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

28

[ FOR { READ ONLY | UPDATE [ OF ім'я таблиці] } ]

де:

**курсор** – ім'я нового курсора;

**BINARY** означає, що вихідні дані повинні повернатися в двійковому форматі замість стандартного ASCII-коду;

**INSENSITIVE** забезпечує незалежність даних, повернених курсором, від інших курсорів або підключень, тобто дані, отримані з використанням курсора, не можуть бути зміненими іншими процесами, наприклад, іншими курсорами;

**SCROLL** забезпечує прокрутку даних, тобто дозволяє багаторазове читання записів;

**CURSOR FOR** *запит* – запит, підсумковий набір якого стає доступним через курсор;

**READ ONLY** дозволяє використовувати курсор тільки для читання даних, цей режим установлений за замовчуванням;

**UPDATE** – курсор використовується для редагування таблиць;

**OF ім'я таблиці** – ім'я таблиці, яка може оновлюватися під час використання курсора.

У поданому нижче прикладі створено транзакцію командою BEGIN і відкрито курсор gazprovid, який буде містити всі записи і всі поля таблиці gazpr:

```
BEGIN;
DECLARE gazprovid CURSOR FOR SELECT * FROM gazpr;
```

Вибірка записів із курсора забезпечується командою FETCH, яка має такий вигляд:

```
FETCH [FORWARD | BACKWARD | RELATIVE] [число | ALL | NEXT | PRIOR]
{IN | FROM} курсор
```

У цьому оголошенні **курсор** – ім'я курсора, з якого відбувається вибірка записів. Курсор завжди встановлений на поточну позицію підсумкового набору. Напрям вибірки визначається ключовими словами FORWARD і BACKWARD, за замовчуванням використовується пряма вибірка (FORWARD). Слово RELATIVE **число** означає зміщення на задану кількість записів відносно поточної позиції курсора. У команді замість RELATIVE може використовуватися ключове слово ABSOLUTE – абсолютне позиціонування, тобто переміщення до заданого запису.

За ключовим словом, який ідентифікує напрям, можна вказувати кількість записів. Допускається вказання конкретного числа записів (у вигляді цілочислової константи) або одного з декількох ключових слів, з яких:

**ALL** означає, що команда повертає всі записи, починаючи з поточної позиції;

**NEXT** (використовується за замовчуванням) – наступний запис від поточної позиції;

**PRIOR** запис, який знаходиться перед поточним.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
29						

Ключові слова **IN** і **FROM** еквівалентні, з них команда повинна мати якесь одне.

Нижче у прикладі вибираються перші чотири записи підсумкового набору, на який посилається курсор **gazprovid**. Напрямок не вказаний, тому за замовчанням використовується ключове слово **FORWARD**. Далі команда **FETCH** з ключовим словом **NEXT** вибирає п'ятий запис, після чого вона з ключовим словом **PRIOR** знову повертається до четвертого запису.

```
FETCH 4 FROM gazprovid;
FETCH NEXT FROM gazprovid;
FETCH PRIOR FROM gazprovid;
```

Число записів може бути як додатнім, так і відємним. При додатному значенні відлік ведеться в напрямку, заданому відповідним параметром (якщо параметр не вказаний, за замовчуванням використовується параметр **FORWARD**).

При від'ємній кількості записів відлік ведеться в напрямку, протилежному заданому. Наприклад, конструкція **FORWARD -2** еквівалентна конструкції **BACKWARD 2**. Якщо кількість запрошеніх записів перевищує кількість записів у базі, команда **FETCH** вибирає всі існуючі записи у вказаному напрямку.

У подальшому прикладі виготовляється курсор **gazprovid**, який повертає дані з таблиці **gazpr**. Наступна команда вибирає перші два його записи:

```
BEGIN;
FETCH FORWARD 2 IN gazprovid;
```

У нижче наведеному прикладі конструкція **BACKWARD -2** також вибирає два записи в прямому напрямку:

```
FETCH BACKWARD - 2 IN gazprovid;
```

А ця команда демонструє вибірку в зворотному напрямку через курсор **gazprovid**:

```
FETCH BACKWARD 3 IN gazprovid;
FETCH PRIOR IN gazprovid; -- на 1 назад
FETCH NEXT IN gazprovid; -- на 1 вперед
FETCH 2 IN gazprovid; -- теж, що NEXT 2
```

Спроба вибірки нуля записів ключовим словом **RELATIVE** видає поточний запис, у деяких ранніх версіях SQL спричиняє помилку

```
ERROR: FETCH/RELATIVE at current position is not supported.
```

Це пов'язано з тим, що відповідно до стандарту SQL92 команда **FETCH RELATIVE FROM** курсор повинна забезпечувати повторну вибірку запису в поточній позиції курсора. У деяких версіях PostgreSQL цей синтаксис не підтримується. Без ключового слова **RELATIVE** число 0 інтерпретується як запит на вибірку всіх записів.

Переміщення курсора до заданого запису виконується командою **MOVE**:

```
MOVE [FORWARD | BACKWARD | RELATIVE] [число | ALL | NEXT | PRIOR] {IN
| FROM} курсор
```

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						30

Як видно з цього оголошення, синтаксис команди **MOVE** дуже близький до синтаксису команди **FETCH**. Втім, команда **MOVE** ніяких записів не повертає і лише переміщає поточну позицію курсора. Зміщення задається цілочисловою константою або ключовими словами **ALL** (переміщення в заданому напрямку на максимально можливу відстань), **NEXT** або **PRIOR**. Нижче у прикладі поточна позиція курсора переміщується на 3 записи вперед:

**MOVE RELATIVE 3 FROM gazprovid;**

Наведемо приклад переміщення поточної позиції курсора:

**MOVE FORWARD 3 IN gazprovid;**

**Закриття курсора** забезпечується командою **CLOSE**. Курсор також автоматично закривається при виході з транзакційного блоку, в якому він знаходиться, при фіксації транзакції командою **COMMIT** або при її відкоті командою **ROLLBACK**.

Команда **CLOSE** має такий вигляд (*курсор* – ім'я курсора, який закривається):

**CLOSE курсор**

Змн.	Арк.	№ докум.	Підпис	Дата

КР.ІП - 25.00.00.000 ПЗ

Арк.

31

## 4 ОСНОВНІ ЕЛЕМЕНТИ МОВИ PL/PGSQL

### 4.1 Структура мови, функції користувача

**PL/pgSQL** є процедурним розширенням мови SQL, тому може використовувати всі її типи даних, оператори і функції. Це підвищує гнучкість використання і швидкодію команд SQL, оскільки в програмному блоці вони виконуються за одну операцію замість звичайної обробки кожної команди.

**Структура мови.** Мова PL/pgSQL відносно проста, кожен її логічно відокремлений фрагмент коду існує у вигляді функції. До певної міри програми подібні до написаних мовою C: всі змінні обов'язково оголошуються перед використанням, функції отримують аргументи при виклику і повертають потрібні значення при закінченні роботи та ін.

Регістр символів в іменах функцій PL/pgSQL не регламентований. У ключових словах і ідентифікаторах допускається використання довільних комбінацій символів верхнього та нижнього регістрів.

**Програмний блок.** При першому виготовленні функції створюється мова програмування командою CREATE LANGUAGE, зазвичайно це plpgsql. Виготовити програмний блок (функцію) можна командою SQL CREATE FUNCTION, у її складі може міститися інструкція OR REPLACE, яка дозволяє редагувати функцію.

Функція являє собою блок, який містить секцію DECLARE – оголошення даних та BEGIN – команди, які виконуються. Закінчується програмний блок словом END. Загальний вигляд програмного блока:

```
CREATE LANGUAGE мова_програмування;
CREATE OR REPLACE FUNCTION ім'я_функції (перелік_типів_аргументів)
RETURNS тип_поверненого_значення
AS $$ DECLARE оголошення даних; BEGIN команди; END; $$
LANGUAGE 'мова_програмування';
```

Коментарі мови PL/pgSQL є двох типів. Коментарі первого типу – однорядкові, починаються з двох дефісів, другого – блокові, беруться у знаки /\* та \*/.

**В оголошенні** задаються імена і типи змінних, кожне оголошення або команда завершується символом крапки з комою. Крім типів даних, визначених мовою SQL, мова PL/pgSQL має ще три додаткові типи, які дозволяють працювати з таблицями та іншими об'єктами БД (вибирати їхні значення), це:

- RECORD – для записів без вказання типів полів;

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						32

- %ROWTYPE – для записів, тип яких відповідає типам полів конкретного об'єкта;
- %TYPE – для змінної, тип якої відповідає типу конкретного поля об'єкта (таблиці).

Ці додаткові типи вигідні тим, що в разі зміни типу даних, наприклад, таблиці, автоматично змінюється й відповідний тип запису чи змінної функції.

Змінні можуть бути ініціалізовані під час оголошення. Аргументи, прийняті функцією, можна спеціально не оголошувати, тоді їхні значення по черзі присвоюються ідентифікаторам, які утворюються автоматично і складаються із знака долара та порядкового номера. Першому аргументу відповідає ідентифікатор \$1, другому – \$2 і т. д. Максимально допустима кількість аргументів дорівнює 16, тому ідентифікатори аргументів лежать у інтервалі від \$1 до \$16. Якщо аргументів є декілька, то з метою уникнення плутанини їх варто замінити псевдонімами – ідентифікаторами внутрішніх змінних функції за допомогою слова ALIAS, наприклад: ідентифікатор\_змінної\_1 ALIAS FOR \$1; ідентифікатор\_змінної\_2 ALIAS FOR \$2; і т. д.

**Присвоєння** значень змінним виконує оператор := подібно до мови Pascal. Мова PL/pgSQL призначена в основному для роботи з таблицями БД, тому має варіант запиту – команду SELECT INTO, яка також дозволяє присвоювати змінним результати його виконання. Ця команда застосовується в основному для запам'ятовування записів (полів) вибірки змінним, оголошених з типами RECORD, %ROWTYPE і %TYPE (для поля). Якщо команда SELECT INTO видає своє значення звичайній змінній, то її тип повинен відповідати типу відповідного запису ( поля) вибірки. Наголосимо на тому, що вона повинна повернати лише одне якесь значення (адже змінна то одна). Синтаксис команди SELECT INTO:

```
SELECT INTO перелік_імен_змінних_перелік_імен_полів
FROM джерело секції_команди_SELECT;
```

**Значення, яке повертає функція**, передає команда RETURN, її наявність у програмному блоці обов'язкова, навіть якщо функція нічого не повертає. Тип цього значення повинен відповідати **типу\_поверненого\_значення**, вказаному при її створенні. Команда RETURN знаходиться наприкінці функції, але вона також часто зустрічається і в командах IF та інших, які здійснюють передачу управління у програмі..

**Виклик функції** (звертання до функції) забезпечує команда:

```
SELECT ім'я_функції(перелік_дійсних_параметрів);
```

## 4.2 Розгалуження, цикли, масиви

**Розгалуження** забезпечують команди:

IF умова THEN команда; END IF;

IF умова THEN команда; ELSE команда; END IF;

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						33

IF умова THEN команда; ELSE IF умова THEN команда; END IF;

Розгалуження можна застосувати для перевірки на відсутність значення змінної, виданого командою SELECT INTO, для цього його слід порівняти з IS NULL.

У PL/pgSQL реалізовано три типи циклів: LOOP, WHILE і FOR.

**Цикл LOOP** виконується доти, поки не буде досягнено ключове слово EXIT. За ключовим словом EXIT може міститися секція WHEN з виразом логічного типу, яка визначає умову виходу із циклу. Без цієї секції матимемо просто зациклення. Цикл LOOP має такий вигляд:

```
LOOP
команди (тіло_циклу);
EXIT [WHEN умова_виходу_з_циклу];
END LOOP;
```

**Цикл WHILE** містить умову свого завершення, він виглядає так:

```
WHILE умова_виходу_з_циклу LOOP
команди (тіло_циклу);
END LOOP;
```

**Цикли FOR** у якості параметра використовує змінну цілого типу, він виглядає так:

```
FOR змінна IN [REVERSE] вираз_1 .. вираз_2 [BY приріст] LOOP
тіло_циклу;
END LOOP;
```

Цей цикл забезпечує виконання свого тіла при кожному значенні змінної (параметра, керівної змінної), межі якої визначаються виразами вираз1 .. вираз2. На початку циклу параметр ініціалізується значенням вираз1 i, якщо не задано приріст, то за замовчуванням він збільшується на 1 після кожного виконання. Якщо в заголовку циклу вказано слово REVERSE, то параметр не збільшується, а зменшується. Керівну змінну циклу не обов'язково оголошувати у блоці DECLARE, якщо вона в програмі більше ніде не використовується.

Тіло циклу FOR може мати команду EXIT [WHEN умова\_виходу\_з\_циклу], яка забезпечує передчасне його завершення (подібно до циклу типу LOOP). Крім того, для продовження циклу з нарощеним значенням параметра тіло може містити команду CONTINUE [WHEN умова\_продовження\_циклу].

**Цикл FOR** можна використовувати для читання запитів, тоді він має такий вигляд:

```
FOR змінна_типу_RECORD_або_%ROWTYPE IN запит LOOP
тіло_циклу;
END LOOP;
```

Тип %ROWTYPE використовується тоді, коли читається одна визначена запитом таблиця.

**Масиви** оголошуються наступним чином:

```
<Ім'я_масиву> <тип_елемента_масиву> array(<початкова_довжина>).
```

Змн.	Арк.	№ докум.	Підпис	Дата

Масиви є динамічними, тобто при виході за межі масиву, масив розширяється до необхідної довжини.

### 4.3 Курсори

**Оголошення курсорної змінної.** Всі доступи до курсора в PL/pgSQL проходять через курсорні змінні, які мають спеціальний тип даних refcursor. Один із способів створення курсорної змінної є просто її оголошення як змінної типу refcursor. Інший спосіб – оголошення курсора, яке має такий синтаксис:

```
ім'я [[ NO ] SCROLL ] CURSOR [( аргументи ) ] FOR запит;
```

FOR можна замінити на IS для сумісності з Oracle. Якщо SCROLL вказаний, то курсор буде здатний до прокрутки назад; якщо вказано NO SCROLL, то зворотна вибірка буде відхиlena, якщо ні одна специфікація не вказана, то дозвіл на читання вибірки назад буде залежати від запиту. Аргументи, перелік яких дається через кому, визначають імена, які повинні бути замінені значеннями параметрів запиту. Фактичні значення для заміни цих імен буде уточнено після відкриття курсора. Приклади:

```
DECLARE
    curs1 REFCURSOR;
    curs2 CURSOR FOR SELECT * FROM джерело;
    curs3 CURSOR (ключ INTEGER) IS SELECT * FROM джерело WHERE поле = ключ;
```

Всі ці три змінні мають тип refcursor, але перша може бути використана з будь-яким запитом, друга має зв'язаний з нею запит і, третій – зв'язаний з нею параметризований запит. Змінна curs1 називається незв'язаною, оскільки вона не прив'язана ні до якого конкретного запиту.

**Відкриття курсора.** Перед використанням курсора для отримання рядків його необхідно відкрити. Це еквівалентно дії команди SQL DECLARE CURSOR. PL/pgSQL має три форми відкриття, дві з яких незв'язані з курсорними змінними, а третя використовує зв'язану змінну курсора.

Змінні, пов'язані з курсором, можуть також використовуватися без явного відкриття курсора, через оператор FOR, описаний у попередньому розділі.

**Відкриття для запиту** (можливе також відкриття для виконання – EXECUTE)

```
OPEN unbound_cursorvar [[ NO ] SCROLL ] FOR запит;
```

Змінну курсора буде відкрито і з урахуванням зазначеного запиту. Курсор не може бути відкритим під час оголошення. Він повинен бути оголошений як незв'язана змінна курсора (тобто, як проста змінна refcursor). Запит повинен бути SELECT або інший, який повертає рядки (наприклад, EXPLAIN). Запит PL/pgSQL такий же, як і інші команди SQL: в PL/pgSQL імена змінних замінюються, і план запиту кешується для можливого повторного

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						35

використання. Варіанти SCROLL і NO SCROLL мають те ж значення, що й для зв'язаного курсора. Наприклад:

```
OPEN curs1 FOR SELECT * FROM foo WHERE key = mykey;
```

**Використання курсорів.** Як тільки курсор відкритий, його даними можна маніпулювати відповідно до потреби. Ці маніпуляції не повинні відбуватися в тій же функції, що відкритий курсор. Внутрішньо, значення refcursor є просто рядок – ім'я так званого порталу, який містить активний запит для курсора. Це ім'я може бути передано з інших змінних refcursor, не порушуючи порталу. Всі порталі неявно закриваються наприкінці запиту, тому значення refcursor може використовуватися тільки до кінця операції.

**Команда FETCH:**

```
FETCH [напрямок {FROM | IN}] курсор INTO ціль;
```

FETCH витягує наступний рядок з курсора в ціль, яка може бути рядком змінних, записом або розділеним комами списком простих змінних, так же як і в SELECT INTO. Як і в SELECT INTO, спеціальна змінна може бути перевірена на наявність – FOUND. Напрямок може бути будь-яким із варіантів дозволених командою SQL FETCH, крім тих, які можуть видавати більше, ніж один рядок, а саме: NEXT, PRIOR, FIRST, LAST, ABSOLUTE count, RELATIVE count, FORWARD, або BACKWARD. Напрямок NEXT опускається. Приклади:

**Переміщення** курсора, команда MOVE:

```
MOVE [напрямок {FROM | IN}] курсор;
```

MOVE переміщує курсор без отримання будь-яких даних. MOVE працює так же, як у команді FETCH, за винятком того, що курсор не повертає рядок. Як і в SELECT INTO та FETCH, відбувається перевірка чи існує рядок, на який потрібно перейти. Напрямок може бути будь-яким із дозволених у команді SQL FETCH, а саме: NEXT, PRIOR, FIRST, LAST, ABSOLUTE count, RELATIVE count, ALL, FORWARD [count | ALL], або BACKWARD [count | ALL]. Напрямок NEXT приймається за замовчуванням. Приклади:

```
MOVE curs1;
MOVE RELATIVE -2 FROM curs4;
MOVE FORWARD 2 FROM curs4;
```

**Редагування таблиці**, команда UPDATE / DELETE WHERE CURRENT OF:

```
UPDATE таблиця SET ... WHERE CURRENT OF курсор;
DELETE FROM таблиця WHERE CURRENT OF курсор;
```

Коли курсор знаходиться на рядку таблиці, її рядки можуть бути оновлені або вилучені за допомогою курсора для ідентифікації рядка.

**Закриття** курсора:      **CLOSE** курсор;

Змн.	Арк.	№ докум.	Підпис	Дата

## 5. ПРАКТИЧНА ЧАСТИНА

### Завдання (Варіант №20)

Задача 1. Спроектувати та утворити базу даних Turyst для автоматизації послуг фірми “Сьорфінг”, яка містить такі дані про фірму “Сьорфінг”:

- 1– найменування маршруту;
- 2– кількість днів;
- 3– дата початку маршруту;
- 4– поточна кількість проданих путівок, шт;
- 5– умови проживання;
- 6– умови харчування;
- 7– ціна за одну путівку, у. о.;
- 8– початкова кількість путівок, шт.

1	2	3	4	5	6	7	8
Австрія (Альпи)	8	01.02.05	8	3* і 5* готелі	3-разове харчування	850	10
Англія (Лондон)	11	05.03.05	15	3* готелі	3-разове харчування	610	20
Гаваї	12	01.02.05	0	3* і 5* готелі	сніданок	565	44
Угорщина (Будапешт)	6	01.02.05	0	3* готелі	сніданок	99	34
Єгипет (Хургада)	7	06.06.05	120	2* і 3* готелі	3-разове харчування	219	126
Польща	7	20.10.05	0	3* і 5* готелі	3-разове харчування	99	224
Польща (Краків)	5	06.06.05	442	3* готелі	сніданок	99	442
Словакія	7	05.03.05	200	3* і 5* готелі	3-разове харчування	255	224
Словакія, Польща	7	01.02.05	0	2* готелі	сніданок	264	100
Тайланд	7	01.04.05	22	3* і 5* готелі	3-разове харчування	700	48
Чехія (Прага)	6	06.06.05	34	3* готелі	сніданок	551	88
Швейцарія	8	01.02.05	47	3* готелі	сніданок	950	124
О. Бали	8	06.06.05	47	2* готелі	сніданок	1190	48
О. Маврикій	8	20.10.05	10	2* і 3* готелі	3-разове харчування	1390	48
Йорданія	7	05.03.05	54	2* готелі	без харчування	330	68
Йорданія	14	20.10.05	12	3* готелі	без харчування	660	48

Утворити таблиці БД. Рекомендації до побудови таблиць:

- 1 - **Маршрут:** номер маршруту, найменування маршруту, кількість днів, кількість путівок, код умов проживання, код умов харчування, ціна за одну путівку;
- 2 - **Облік реалізації:** номер маршруту, дата початку маршруту, поточна кількість проданих путівок;
- 3 - **Проживання:** код умов проживання, умови проживання;
- 4 - **Харчування:** код умов харчування, умови харчування.

Задача 2. Підготувати і занести в таблиці контрольні дані.

Задача 3. Виготовити запит про розподіл кількості путівок з певними умовами проживання по умовах харчування. Навпроти умов проживання встановити кількість путівок для кожного виду харчування. Передбачити підсумки по вертикалі і по горизонталі.

Задача 4. Утворити запит про помісячну сумарну виручку за путівки в поточному році. Виручка за путівки дорівнює її ціні, помножений на кількість проданих.

Задача 5. Виготовити запит про “тарячі” путівки, тобто такі, що ще незакуплені, а до початку маршруту залишилося менше 6 днів. У звіт включити дані про номер і називу

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
37						

маршруту, ціну за одну путівку та кількість путівок, які ще залишилися. До заголовка звіту додати слова “станом на ” і встановити поточну дату.

Задача 6. Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про нові маршрути і редагування кількості проданих путівок.

Задача 7. Виготовити запитки з відомостями про путівки фірми. Забезпечити відповідний напис для зимових путівок, тобто там, де дата початку маршруту припадає на місяці від грудня до лютого включно.

Задача 8. Скласти програму для видачі звіту за умовою задачі 5, використати курсор.

### 5.1 Завдання 1 (створення таблиць, обмежень)

Спроектувати та утворити базу даних Turyst для автоматизації послуг фірми “Сьюрфінг” на основі таблиці 2.20, яка містить такі дані про фірму “Сьюрфінг”:

- 1– найменування маршруту;
- 2– кількість днів;
- 3– дата початку маршруту;
- 4– поточна кількість проданих путівок, шт;
- 5– умови проживання;
- 6– умови харчування;
- 7– ціна за одну путівку, у. о.;
- 8– початкова кількість путівок, шт.

#### Розв'язок

--Створення таблиці маршрут

```
CREATE TABLE marshrut (nom INTEGER, name CHARACTER(40), kil_d FLOAT,  
kil_p INTEGER, k_u_p INTEGER, k_u_h INTEGER, cina FLOAT);
```

--Створення таблиці облік реалізації

```
CREATE TABLE oblik_realizaciy (nom INTEGER, data DATE, kil_p_p INTEGER);
```

--Створення таблиці проживання

```
CREATE TABLE prozgyvannya (k_u_p INTEGER, u_p CHARACTER (80));
```

--Створення таблиці харчування

```
CREATE TABLE harchuvannya (k_u_h INTEGER, u_h CHARACTER(80));
```

-- Додаємо обмеження до таблиць

```
ALTER TABLE marshrut ADD PRIMARY KEY (nom) ;  
ALTER TABLE oblik_realizaciy ADD FOREIGN KEY (nom) REFERENCES  
marshrut(nom) ;  
ALTER TABLE marshrut ADD PRIMARY KEY (k_u_p) ;  
ALTER TABLE marshrut ADD PRIMARY KEY (k_u_h) ;  
ALTER TABLE prozgyvannya ADD CONSTRAINT cs_marshrut FOREIGN KEY (k_u_p)  
REFERENCES marshrut(k_u_p) ;  
ALTER TABLE harchuvannya ADD CONSTRAINT cs_marshrut FOREIGN KEY (k_u_h)  
REFERENCES marshrut(k_u_h) ;  
ALTER TABLE marshrut ADD CONSTRAINT cs_marshrut CHECK (kil_p>0) ;  
ALTER TABLE marshrut ADD CONSTRAINT cs_marshrut CHECK (kil_d>0) ;  
ALTER TABLE marshrut ADD CONSTRAINT cs_marshrut CHECK (cina>0) ;
```

Змн.	Арк.	№ докум.	Підпис	Дата

КР.ІІ - 25.00.00.000 ПЗ

Арк.

38

```
ALTER TABLE oblik_realizaciy ADD CONSTRAINT cs_oblik_realizaciy CHECK  
(kil_p_p>5);
```

-- Створення послідовності `setnom` і додавання обмеження `DEFAULT` до поля `nom` таблиці `marshrut`, використовуючи дану послідовність:

```
CREATE SEQUENCE setnom minvalue 0;  
ALTER TABLE marshrut ALTER nom SET DEFAULT nextval('setnom');
```

**Результати виконання:**

Properties		
Table	Owner	Comment
harchuvannya	postgres	
marshrut	postgres	
oblik_realizaciy	postgres	
prozgyvannya	postgres	

## 5.2 Завдання 2 (заповнення таблиць даними)

Підготувати і занести в таблиці контрольні дані, відредагувати, якщо потрібно

### Розв'язок задачі

-- Вставляємо контрольні дані в таблицю 'Маршрут'

```
INSERT INTO marshrut VALUES(1, 'Івано-Франківськ-Яремче', 2.5, 20, 1, 2, 230);  
INSERT INTO marshrut VALUES(2, 'Київ-Прага', 10, 10, 1, 2, 850.5);  
INSERT INTO marshrut VALUES(3, 'Львів-Одеса', 7, 15, 1, 2, 400);  
INSERT INTO marshrut VALUES(4, 'Київ-Лондон', 25, 10, 1, 2, 4000);  
INSERT INTO marshrut VALUES(5, 'Івано-Франківськ-Крим', 7, 40, 1, 2, 680);
```

-- Вставляємо контрольні дані в таблицю 'Облік реалізацій'

```
INSERT INTO oblik_realizaciy VALUES(1, '20.10.2012', 20);  
INSERT INTO oblik_realizaciy VALUES(2, '16.12.2012', 9);  
INSERT INTO oblik_realizaciy VALUES(3, '5.11.2012', 12);  
INSERT INTO oblik_realizaciy VALUES(4, '5.05.2012', 8);  
INSERT INTO oblik_realizaciy VALUES(5, '18.12.2011', 37);  
INSERT INTO oblik_realizaciy VALUES(1, '10.10.2012', 14);  
INSERT INTO oblik_realizaciy VALUES(1, '22.06.2011', 19);  
INSERT INTO oblik_realizaciy VALUES(3, '05.10.2009', 14);  
INSERT INTO oblik_realizaciy VALUES(2, '02.09.2012', 10);  
INSERT INTO oblik_realizaciy VALUES(3, '06.01.2013', 18);  
INSERT INTO oblik_realizaciy VALUES(1, '11.02.2012', 10);  
INSERT INTO oblik_realizaciy VALUES(4, '10.12.2012', 8);  
INSERT INTO oblik_realizaciy VALUES(5, '10.12.2012', 37);
```

-- Вставляємо контрольні дані в таблицю 'Проживання'

```
INSERT INTO prozgyvannya VALUES(1, '3* готелі');  
INSERT INTO prozgyvannya VALUES(1, '3* і 5* готелі');
```

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						39

```
INSERT INTO prozgyvannya VALUES(1, '2* і 3* готелі');
```

-- Вставляємо контрольні дані в таблицю ‘Харчування’

```
INSERT INTO harchuvannya VALUES(2, '3-разове харчування');  
INSERT INTO harchuvannya VALUES(2, 'сніданок');  
INSERT INTO harchuvannya VALUES(2, 'без харчування');
```

-- Редагуємо контрольні дані в таблиці ‘Облік реалізації’

```
UPDATE oblik_realizaciy SET kil_p_p = 10 WHERE nom = 4;  
UPDATE oblik_realizaciy SET kil_p_p = 39 WHERE nom = 5;
```

-- Редагуємо контрольні дані в таблиці ‘Проживання’

```
UPDATE prozgyvannya SET k_u_p = 2 WHERE u_p = '3* готелі';  
UPDATE prozgyvannya SET k_u_p = 3 WHERE u_p = '3* і 5* готелі';
```

-- Редагуємо контрольні дані в таблиці ‘Харчування’

```
UPDATE harchuvannya SET k_u_h = 4 WHERE u_h = '3-разове харчування';  
UPDATE harchuvannya SET k_u_h = 5 WHERE u_h = 'сніданок';  
UPDATE harchuvannya SET k_u_h = 6 WHERE u_h = 'без харчування';
```

-- Редагуємо контрольні дані в таблиці ‘Маршрут’

```
UPDATE marshrut SET k_u_p = 1 WHERE nom = 1;  
UPDATE marshrut SET k_u_p = 2 WHERE nom = 2;  
UPDATE marshrut SET k_u_p = 3 WHERE nom = 3;  
UPDATE marshrut SET k_u_p = 1 WHERE nom = 4;  
UPDATE marshrut SET k_u_p = 3 WHERE nom = 5;  
UPDATE marshrut SET k_u_h = 4 WHERE nom = 1;  
UPDATE marshrut SET k_u_h = 5 WHERE nom = 2;  
UPDATE marshrut SET k_u_h = 6 WHERE nom = 3;  
UPDATE marshrut SET k_u_h = 5 WHERE nom = 4;  
UPDATE marshrut SET k_u_h = 4 WHERE nom = 5;  
UPDATE marshrut SET cina = 980 WHERE nom = 4;
```

### Результати виконання завдання

nom integer	name character(40)	kil_d double precis	kil_p integer	k_u_p integer	k_u_h integer	cina double precis
1	Івано-Франківськ-Яремче	2.5	20	1	4	230
2	Київ-Прага	10	10	2	5	850.5
3	Львів-Одеса	7	15	3	6	400
4	Івано-Франківськ-Крим	7	40	3	4	680
5	Київ-Лондон	25	10	1	5	980

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

40

	nom integer	data date	kil_p_p integer
1	1	2012-10-20	20
2	2	2012-12-16	9
3	3	2012-11-05	12
4	1	2012-10-10	14
5	1	2011-06-22	19
6	3	2009-10-05	14
7	2	2012-09-02	10
8	3	2013-01-06	18
9	1	2012-02-11	10
10	4	2012-05-05	10
11	4	2012-12-10	10
12	5	2011-12-18	39
13	5	2012-12-10	39

	k_u_p integer	u_p character(80)
1	1	2* і 3* хотели
2	2	3* хотели
3	3	3* і 5* хотели

	k_u_h integer	u_h character(80)
1	4	3-разове харчування
2	5	сніданок
3	6	без харчування

### 5.3 Завдання 4 (Запити)

Утворити запит про помісячну сумарну виручку за путівки в поточному році. Виручка за путівки дорівнює її ціні, помноженій на кількість проданих.

#### Розв'язок задачі

```
SELECT name, sum(marshrut.cina*oblik_realizaciy.kil_p_p),
extract (month FROM data) AS mis, extract (year FROM data) AS rik FROM
oblik_realizaciy
INNER JOIN marshrut ON (marshrut.nom=oblik_realizaciy.nom)
WHERE extract (year FROM data)=extract (year FROM current_date)
GROUP BY name, mis, rik
ORDER BY name;
```

#### Результати запиту

Змн.	Арк.	№ докум.	Підпис	Дата	Арк.
					41

КР.ІІ - 25.00.00.000 ПЗ

	<b>name character(40)</b>	<b>sum double precision</b>	<b>mis double precision</b>	<b>rik double precision</b>
<b>1</b>	Івано-Франківськ-Яремче	7820	10	2012
<b>2</b>	Київ-Прага	8505	9	2012
<b>3</b>	Київ-Прага	7654.5	12	2012
<b>4</b>	Львів-Одеса	4800	11	2012

#### 5.4 Завдання 6 (Запити)

Виготовити запити для перегляду вмісту таблиць з даними та доповнення БД відомостями про нові маршрути і редагування кількості проданих путівок.

#### Розв'язок задачі

```
SELECT *FROM marshrut;
SELECT *FROM oblik_realizaciy;
SELECT *FROM prozgyvannya;
SELECT *FROM harchuvannya;
```

	<b>nom integer</b>	<b>name character(40)</b>	<b>kil_d double precis</b>	<b>kil_p integer</b>	<b>k_u_p integer</b>	<b>k_u_h integer</b>	<b>cina double precis</b>
<b>1</b>	1	Івано-Франківськ-Яремче	2.5	20	1	4	230
<b>2</b>	2	Київ-Прага	10	10	2	5	850.5
<b>3</b>	3	Львів-Одеса	7	15	3	6	400
<b>4</b>	5	Івано-Франківськ-Крим	7	40	3	4	680
<b>5</b>	4	Київ-Лондон	25	10	1	5	980

	<b>nom integer</b>	<b>data date</b>	<b>kil_p_p integer</b>
<b>1</b>	1	2012-10-20	20
<b>2</b>	2	2012-12-16	9
<b>3</b>	3	2012-11-05	12
<b>4</b>	1	2012-10-10	14
<b>5</b>	1	2011-06-22	19
<b>6</b>	3	2009-10-05	14
<b>7</b>	2	2012-09-02	10
<b>8</b>	3	2013-01-06	18
<b>9</b>	1	2012-02-11	10
<b>10</b>	4	2012-05-05	10
<b>11</b>	4	2012-12-10	10
<b>12</b>	5	2011-12-18	39
<b>13</b>	5	2012-12-10	39

	<b>k_u_p integer</b>	<b>u_p character(80)</b>
<b>1</b>	1	3* хотелі
<b>2</b>	1	3* і 5* хотелі
<b>3</b>	1	2* і 3* хотелі

  

	<b>k_u_h integer</b>	<b>u_h character(80)</b>
<b>1</b>	2	3-разове харчування
<b>2</b>	2	сніданок
<b>3</b>	2	без харчування

```
SELECT name, kil_p_p
FROM marshrut, oblik_realizaciy
WHERE oblik_realizaciy.nom=marshrut.nom AND oblik_realizaciy.nom=4;
```

	<b>name character(40)</b>	<b>kil_p_p integer</b>
<b>1</b>	Київ-Лондон	10

```
SELECT name, kil_p_p
FROM marshrut, oblik_realizaciy
WHERE oblik_realizaciy.nom=marshrut.nom AND oblik_realizaciy.nom=5;
```

	<b>name character(40)</b>	<b>kil_p_p integer</b>
<b>1</b>	Івано-Франківськ-Крим	39

--При виготовленні запиту, використовуємо JOIN для об'єднання таблиць

```
SELECT * FROM oblik_realizaciy
```

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

42

```

INNER JOIN marshrut ON marshrut.nom=oblik_realizaciy.nom
INNER JOIN prozgyvannya ON marshrut.k_u_p=prozgyvannya.k_u_p
INNER JOIN harchuvannya ON marshrut.k_u_h=harchuvannya.k_u_h;

```

### Результати запиту

	nom integer	data date	kil_p_p integer	nom integer	name character(40)	kil_d double integer	kil_p integer	k_u_p integer	k_u_h integer	cina dou integer	k_u_p character(80)	u_p character(80)	k_u_h integer	u_h character(80)	
1	1	2012-10-20	20	1	Івано-Франківськ-Яремче	2.5	20	1	4	230	1	2* і 3* хотелі	4	3-разове харчування	
2	2	2012-12-16	9	2	Київ-Прага		10	10	2	5	850	2	3* хотелі	5	сніданок
3	3	2012-11-05	12	3	Львів-Одеса		7	15	3	6	400	3	3* і 5* хотелі	6	без харчування
4	4	2012-05-05	10	4	Київ-Лондон		25	10	1	5	400	1	2* і 3* хотелі	5	сніданок
5	5	2011-12-18	39	5	Івано-Франківськ-Крим	7	40	3	4	680	3	3* і 5* хотелі	4	3-разове харчування	
6	1	2012-10-10	14	1	Івано-Франківськ-Яремче	2.5	20	1	4	230	1	2* і 3* хотелі	4	3-разове харчування	
7	1	2011-06-22	19	1	Івано-Франківськ-Яремче	2.5	20	1	4	230	1	2* і 3* хотелі	4	3-разове харчування	
8	3	2009-10-05	14	3	Львів-Одеса		7	15	3	6	400	3	3* і 5* хотелі	6	без харчування
9	2	2012-09-02	10	2	Київ-Прага		10	10	2	5	850	2	3* хотелі	5	сніданок
10	3	2013-01-06	18	3	Львів-Одеса		7	15	3	6	400	3	3* і 5* хотелі	6	без харчування
11	1	2012-02-11	10	1	Івано-Франківськ-Яремче	2.5	20	1	4	230	1	2* і 3* хотелі	4	3-разове харчування	

### 5.5 Задача 7 (запити)

Виготовити запит з відомостями про путівки фірми. Забезпечити відповідний напис для зимових путівок, тобто там, де дата початку маршруту припадає на місяці від грудня до лютого включно.

### Розв'язок задачі

```

SELECT *, CASE WHEN extract( month from oblik_realizaciy.data) = 12 OR
                extract( month from oblik_realizaciy.data) = 1 OR
                extract( month from oblik_realizaciy.data) = 2 THEN 'Зимові путівки!' END
                AS zima FROM oblik_realizaciy
INNER JOIN marshrut ON marshrut.nom=oblik_realizaciy.nom
INNER JOIN prozgyvannya ON marshrut.k_u_p = prozgyvannya.k_u_p
INNER JOIN harchuvannya ON marshrut.k_u_h = harchuvannya.k_u_h
WHERE extract (month from oblik_realizaciy.data) = 12 OR extract( month
from oblik_realizaciy.data) = 1 OR
extract( month from oblik_realizaciy.data) = 2;

```

### Результати запиту:

	nom integer	data date	kil_p_p integer	nom integer	name character(40)	kil_d double integer	kil_p integer	k_u_p integer	k_u_h integer	cina dou integer	k_u_p character(80)	u_p character(80)	k_u_h integer	u_h character(80)	zima text	
1	5	2012-39	5	5	Івано-Франківськ-Крим	7	40	3	4	680	3	3* і 5* хотелі	4	3-разове харчування	Зимо	
2	5	2011-39	5	5	Івано-Франківськ-Крим	7	40	3	4	680	3	3* і 5* хотелі	4	3-разове харчування	Зимо	
3	1	2012-10	1	1	Івано-Франківськ-Яремче	2.5	20	1	4	230	1	2* і 3* хотелі	4	3-разове харчування	Зимо	
4	2	2012-9	2	2	Київ-Прага		10	10	2	5	850	2	3* хотелі	5	сніданок	Зимо
5	4	2012-10	4	4	Київ-Лондон		25	10	1	5	980	1	2* і 3* хотелі	5	сніданок	Зимо
6	3	2013-18	3	3	Львів-Одеса		7	15	3	6	400	3	3* і 5* хотелі	6	без харчування	Зимо

### 5.6 Завдання 3 (Звіти, мова PLpgSQL)

Виготовити звіт про розподіл кількості путівок з певними умовами проживання по умовах харчування. Навпроти умов проживання встановити кількість путівок для кожного виду харчування. Передбачити підсумки по вертикалі і по горизонталі.

### Розв'язок задачі

															Арк.
Змн.	Арк.	№ докум.		Підпис	Дата										43

КР.П - 25.00.00.000 ПЗ

```

CREATE OR REPLACE FUNCTION zvit() RETURNS text AS
$$
DECLARE
--оголошення запису
rec RECORD;
--оголошення змінної
kod_u_p prozgyvannya.k_u_p %TYPE = 0;
all_text text := ' ';

BEGIN
all_text:=repeat(' ', 20)||'Звіт'||E'\n';
FOR rec IN SELECT marshrut.kil_p, marshrut.k_u_p AS m_k_u_p, marshrut.k_u_h AS m_k_u_h,
    prozgyvannya.k_u_p AS p_k_u_p, harchuvannya.k_u_h AS h_k_u_h,
    prozgyvannya.u_p, harchuvannya.u_h   FROM marshrut
    INNER JOIN prozgyvannya ON marshrut.k_u_p = prozgyvannya.k_u_p
    INNER JOIN harchuvannya ON marshrut.k_u_h = harchuvannya.k_u_h
    ORDER BY marshrut.k_u_p
LOOP
IF kod_u_p <> rec.m_k_u_p
THEN
    all_text := all_text||rec.u_p||E':\n'||repeat(' ',5)||rpad(rec.u_h,30, '- ')
    ||TO_CHAR(rec.kil_p, '99')||E'\n';
ELSE
    all_text := all_text||repeat(' ',5)||rpad(rec.u_h,30, '- ')
    ||TO_CHAR(rec.kil_p, '99')||E'\n';
END IF;
kod_u_p := rec.m_k_u_p;
END LOOP;
RETURN all_text;
END;
$$
language plpgsql;
select zvit();

```

### Результат задачі

zvit text											
	<p style="text-align: center;">Звіт</p> <table> <tbody> <tr> <td>2* і 3* готелі:</td> <td>3-разове харчування----- 20</td> </tr> <tr> <td></td> <td>сніданок----- 10</td> </tr> <tr> <td>3* готелі:</td> <td>сніданок----- 10</td> </tr> <tr> <td>3* і 5* готелі:</td> <td>без харчування----- 15</td> </tr> <tr> <td></td> <td>3-разове харчування----- 40</td> </tr> </tbody> </table>	2* і 3* готелі:	3-разове харчування----- 20		сніданок----- 10	3* готелі:	сніданок----- 10	3* і 5* готелі:	без харчування----- 15		3-разове харчування----- 40
2* і 3* готелі:	3-разове харчування----- 20										
	сніданок----- 10										
3* готелі:	сніданок----- 10										
3* і 5* готелі:	без харчування----- 15										
	3-разове харчування----- 40										

### 5.7 Завдання 5,8 (Звіти, мова PLpgSQL)

Виготовити звіт про “гарячі” путівки, тобто такі, що ще незакуплені, а до початку маршруту залишилося менше 6 днів. У звіт включити дані про номер і називу маршруту, ціну за одну путівку та кількість путівок, які ще залишилися. До заголовка звіту додати слова “станом на ” і встановити поточну дату.

Скласти програму для видачі звіту за умовою цього завдання, використати курсор.

### Розв’язок задачі

```

CREATE OR REPLACE FUNCTION zvit() RETURNS text AS
$$
DECLARE
--оголошення змінних, масиву, курсору
nomer marshrut.nom %TYPE; imya marshrut.name %TYPE; cina marshrut.cina %TYPE;
den INTEGER; mis INTEGER; rik INTEGER; ms CHARACTER(9) ARRAY[12];
slovo text; sum_1 marshrut.kil_p %TYPE;

```

Змн.	Арк.	№ докум.	Підпис	Дата

КР.ІІ - 25.00.00.000 ПЗ

Арк.

44

```

kursor CURSOR FOR SELECT marshrut.nom, marshrut.name, marshrut.cina ,
sum(marshrut.kil_p-oblik_realizaciy.kil_p_p) AS sum FROM oblik_realizaciy
    INNER JOIN marshrut ON (marshrut.nom=oblik_realizaciy.nom)
    WHERE extract(day FROM data)< extract(day FROM current_date) + 6
    AND extract (month FROM data)=extract (month FROM current_date)
    AND extract (year FROM data)=extract (year FROM current_date)
    GROUP BY marshrut.nom,marshrut.name,marshrut.cina
    ORDER BY name;
BEGIN
--заповнення масиву даними
den:=EXTRACT(DAY FROM CURRENT_DATE);
mis:=EXTRACT(MONTH FROM CURRENT_DATE);
rik:=EXTRACT(YEAR FROM CURRENT_DATE);
ms[1]:='січня';ms[2]:='лютого';ms[3]:='березня';ms[4]:='квітня';ms[5]:='травня';ms[6]:='червня';
ms[7]:='липня';ms[8]:='серпня';ms[9]:='вересня';ms[10]:='жовтня';ms[11]:='листопада';ms[12]:='грудня';
SELECT INTO nomer nom, marshrut.cina cina FROM marshrut;
slovo:='Звіт'||E'\n';
slovo:=slovo ||'про "гарячі" путівки, що ще не закуплені, а до початку маршруту
залишилося менше 6 днів'||E'\n';
slovo:=slovo ||' станом на: '||TO_CHAR(den,'99')||'
'||ms[mis]||' '|| TO_CHAR(rik,'9999')||E'\n\n';
slovo:=slovo ||' Номер'||' '||' Путівка'||' '||'Ціна за 1
путівку'||' '||'Кількість'||E'\n';
slovo:=slovo ||' -----
-----'||E'\n';
open kursor;
LOOP
FETCH kursor INTO nomer,imya,cina,sum_1;
exit when not found;
slovo:=slovo ||' '||nomer||' '||rpad(imya,30,' ')||cina||'
'||TO_CHAR(sum_1,'9')||E'\n';
END LOOP;
CLOSE kursor;
RETURN slovo;
END;
$$
language plpgsql;
select zvit();

```

### Результат задачі:

zvit text					
1	Звіт про "гарячі" путівки, що ще не закуплені, а до початку маршруту залишилося менше 6 днів станом на: 6 грудня 2012				
	Номер	Путівка	Ціна за 1 путівку:	Кількість	
	5	Івано-Франківськ-Крим	680	3	
	4	Київ-Лондон	980	2	

Змн.	Арк.	№ докум.	Підпис	Дата

КР.П - 25.00.00.000 ПЗ

Арк.

45

## **6. ВИСНОВКИ**

В даній курсовій роботі було розглянуто роботу з реляційними базами даних на прикладі PostgreSQL. Було реалізовано завдання по створенню бази даних для обробки інформації в туристичній області, зокрема отримання відомостей про путівки фірми «Сьорфінг», тому для цього використовувалась мова запитів SQL і мова програмування pl/pgSQL.

Система управління базами даних PostgreSQL, що є однією з найрозвиненіших в своїй категорії, дозволяє повноцінну реалізацію баз даних на основі SQL, забезпечує всі стандарти SQL, і крім того дозволяє використання власних додаткових можливостей.

Отже, можна зробити висновок, що SQL - це високорівнева мова запитів, призначена для роботи з базами даних. Вона дозволяє модифікувати дані, складати і виконувати запити, виводити результати у вигляді звітів.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.П - 25.00.00.000 ПЗ	Арк.
						46

## 7. СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. <http://www.postgresql.org/docs/>
2. <http://www.postgresql.org/docs/9.2/interactive/index.html>
3. <http://postgresql.ru.net/>
4. Вовк Р.Б. Конспект лекцій. “Бази даних” . - 2020
5. В.В. Кириллов Основы проектирования реляционных баз данных. Учебное пособие. - СПб.: ИТМО, 1994. - 90 с.
6. М. Мейер Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
7. PostgreSQL Reference Manual - Volume 1: SQL Language Reference — The PostgreSQL Global Development Group, 2007.
8. Уорсли Дж., Дрейк Дж. PostgreSQL. Для профессионалов. – СПб.: Питер, 2003. – 496с.

Змн.	Арк.	№ докум.	Підпис	Дата	КР.ПП - 25.00.00.000 ПЗ	Арк.
						47

## **РЕКОМЕНДОВАНА ЛІТЕРАТУРА**

1. <http://www.postgresql.org/docs/>
2. <http://www.postgresql.org/docs/9.2/interactive/index.html>
3. <http://postgresql.ru.net/>
4. <http://php.net/manual/ru/bookpgsql.php>
5. <http://www.scribd.com/doc/4929964/PostgreSQL-1>
6. [http://citforum.ru/database/advanced\\_intro/82.shtml](http://citforum.ru/database/advanced_intro/82.shtml)
7. <http://works.doklad.ru/view/oOcuUsA7fp4.html>
8. <http://contact.com.ua/PostgreSQL>
9. Вовк Р.Б. Конспект лекцій. “Бази даних” (електронний варіант)
10. Уорсли Дж., Дрейк Дж. PostgreSQL. Для професіоналов. – СПб.: Питер, 2003. – 496с.
11. М. Мейер Теория реляционных баз данных. - М.: Мир, 1987. - 608 с.
12. PostgreSQL Reference Manual - Volume 1: SQL Language Reference — The PostgreSQL Global Development Group, 2007.
13. <http://java2s.com/Code/PostgreSQL/CatalogPostgreSQL.htm>
14. <http://sqlformat.appspot.com/>
15. <http://habrahabr.ru/post/47031/>
16. И. Ф. Астахова, А. П. Толстобров, В. М. Мельников. SQL в примерах и задачах. Учебное пособие 2001.
17. Максимов Е.М. Структурированный язык запросов – SQL
18. Дейт К. Введение в системы баз данных. К.; М.; Спб; Издат. Дом «Вильямс». 2000.