

Домашні Роботи з Бази даних та інформаційних систем

Захаров Дмитро

28 листопада, 2024

Зміст

1	Домашня Робота 1: Підготовка	2
2	Домашня Робота 2: Вибірка з умовами, сортування, групування	3
3	Домашня Робота 3: Перетворення типів	7
4	Домашня Робота 4: Предикати	9
5	Домашня Робота 5: Поєднання таблиць	11
6	Домашня Робота 6: Підзапити. Псевдоніми	14
7	Домашня Робота 7: Регулярні вирази	17
8	Домашня Робота 8: Створення, редагування, видалення	20

1 Домашня Робота 1: Підготовка

Умова Задачі 1.1. Потрібно:

1. Встановити обладнання за моєю інструкцією з практики 1.
2. Ознайомитися з елементами інтерфейсу.

Відповідь. Це завдання суто по установці, тому відповідь на неї не наводжу (проте, обладнання було встановлене!).

2 Домашня Робота 2: Вибірка з умовами, сортування, групування

Умова Задачі 2.1. Є таблиця employee

l_name	f_name	age	sex	status	dpt	position	salary
Mashko	Oleg	22	m	single	IT	Admin	20000
Petrenko	Zoya	24	m	married	IT	Programmer	10000
Vovk	Zahar	32	m	single	Sales	Admin	8000
Kononov	Oleg	26	f	single	Sales	Sales	8500
Galenda	Anna	26	m	single	IT	Programmer	25000
Osokor	Galina	18	f	married	IT	QA	18000

1. Вибрати прізвища та посади всіх, у кого ім'я Oleg, впорядкувати за прізвищами за алфавітом
2. Вибрати імена, прізвища, стать (sex) та вік усіх, хто не є програмістами, впорядкувати за віком в порядку зменшення.
3. Підрахувати та вивести середню зарплату усіх співробітників, яким не 26 років.
4. Вибрати прізвища, відділи (dpt) та посади всіх співробітників із зарплатами понад 9000, впорядкувати за прізвищами.
5. Вибрати всіх, хто не має сім'ї (single) і не адмін.
6. Вибрати відділи (dpt), у якому працюють адміни.
7. Написати, що виведе наступний запит: `SELECT f_name, l_name, age FROM employee WHERE status='single' ORDER BY 2 DESC`
8. Написати, що виведе наступний запит: `SELECT l_name, sex, status FROM employee ORDER BY status, sex`
9. Підрахувати та вивести максимальній вік адмінів.
10. Вивести список співробітників, що мають зарплату, вищу за середню

Відповідь. Перед виконанням усіх завдань, створимо задану таблицю:

```
1  -- Specifying the table schema
2  CREATE TABLE employee (
3      l_name VARCHAR(50),
4      f_name VARCHAR(50),
5      age INT,
6      sex CHAR(1),
7      status VARCHAR(20),
8      dpt VARCHAR(20),
9      position VARCHAR(50),
10     salary INT
11 );
12
```

```
13 -- Inserting the data
14 INSERT INTO employee (l_name, f_name, age, sex, status, dpt,
    ↳ position, salary)
15 VALUES
16 ('Mashko', 'Oleg', 22, 'm', 'single', 'IT', 'Admin', 20000),
17 ('Petrenko', 'Zoya', 24, 'm', 'married', 'IT', 'Programmer', 10000),
18 ('Vovk', 'Zahar', 32, 'm', 'single', 'Sales', 'Admin', 8000),
19 ('Kononov', 'Oleg', 26, 'f', 'single', 'Sales', 'Sales', 8500),
20 ('Galenda', 'Anna', 26, 'm', 'single', 'IT', 'Programmer', 25000),
21 ('Osokor', 'Galina', 18, 'f', 'married', 'IT', 'QA', 18000);
```

Питання 1. Використовуємо наступний запит:

```
1 SELECT l_name, position
2 FROM employee
3 WHERE f_name = 'Oleg'
4 ORDER BY l_name;
```

Як і просить завдання, ми вибираємо значення колонок l_name та position з таблиці employee, де значення колонки f_name дорівнює Oleg. Для сортування за прізвищами використовуємо ORDER BY l_name. Результат виконання запиту:

l_name	position
Kononov	Sales
Mashko	Admin

Питання 2. Використовуємо наступний запит:

```
1 SELECT f_name, l_name, sex, age
2 FROM employee
3 WHERE position != 'Programmer'
4 ORDER BY age DESC;
```

Як результат, отримуємо:

f_name	l_name	sex	age
Zahar	Vovk	m	32
Oleg	Kononov	f	26
Oleg	Mashko	m	22
Galina	Osokor	f	18

Питання 3. Використовуємо наступний запит:

```
1 SELECT AVG(salary) AS average_salary
2 FROM employee
3 WHERE age != 26;
```

Зарплати співробітників, яким не 26 років: 20000, 10000, 8000, 18000, тому середня зарплата буде дорівнювати 14000.

Питання 4. Використовуємо наступний запит:

```
1 SELECT l_name, dpt, position
2 FROM employee
3 WHERE salary > 9000
4 ORDER BY l_name;
```

Результат виконання запиту:

l_name	dpt	position
Galenda	IT	Programmer
Mashko	IT	Admin
Osokor	IT	QA
Petrenko	IT	Programmer

Питання 5. Використовуємо наступний запит:

```
1 SELECT *
2 FROM employee
3 WHERE status = 'single' AND position != 'Admin';
```

Цей запис поверне усі колонки робітників Petrenko Zoya та Galenda Anna.

Питання 6. Використовуємо наступний запит:

```
1 SELECT DISTINCT dpt
2 FROM employee
3 WHERE position = 'Admin';
```

Результатом буде одна колонка з рядками IT та Sales.

Питання 7. SELECT f_name, l_name, age FROM employee WHERE status='single' ORDER BY 2 DESC вибере імена, прізвища та вік усіх співробітників, які не одружені, впорядковані за прізвищами (друга колонка) у зворотньому порядку. Виведе наступне:

f_name	l_name	age
Zahar	Vovk	32
Oleg	Mashko	22
Oleg	Kononov	26
Anna	Galenda	26

Питання 8. SELECT l_name, sex, status FROM employee ORDER BY status, sex вибере прізвища, стать та статус усіх співробітників, впорядкованих спочатку за статусом, а потім за статтю. Результат:

l_name	sex	status
Osokor	f	married
Petrenko	m	married
Kononov	f	single
Galenda	m	single
Vovk	m	single
Mashko	m	single

Питання 9. Використовуємо наступний запит:

```
1 SELECT MAX(age) AS max_age
2 FROM employee
3 WHERE position = 'Admin';
```

Максимальний вік адмінів дорівнює 32 роки (співробітник Zahar Vovk).

Питання 10. Використовуємо наступний запит:

```
1 SELECT *
2 FROM employee
3 WHERE salary > (SELECT AVG(salary) FROM employee);
```

На вихід буде виведено усі колонки співробітників Mashko Oleg, Galenda Anna, Osokor Galina.

3 Домашня Робота 3: Перетворення типів

Умова Задачі 3.1. Таблиця та сама, що і в попередньому завданні. Потрібно вивести результат наступних запитів:

1. `SELECT l_name, f_name, sex FROM employee WHERE dpt=0.`
2. `SELECT l_name, f_name, sex FROM employee WHERE 'dpt'=0.`
3. `SELECT f_name, sex, age FROM employee WHERE salary>'10000'.`
4. `SELECT CAST ('20160911' AS date).`
5. `SELECT l_name, position, dpt FROM employee WHERE salary>CONCAT(age, '7000').`
6. `SELECT l_name, f_name, position FROM employee WHERE 'salary' >= 0.`
7. Написати запит, який виведе прізвища та імена кожного працівника (одним полем), а також вік, стать, статус, посаду у порядку зростання віку.

Відповідь.

Питання 1. Оскільки ми задали `dpt` як `VARCHAR`, то в більшості SQL-систем порівняння з нулем видасть помилку. Зокрема, мій компілятор видає наступну помилку:

```
Conversion failed when converting the varchar value 'IT' to data type int.
```

Питання 2. У цьому запиті, `dpt` не є посиланням на стовпець таблиці, а просто текстовим рядком зі значенням `'dpt'`. SQL-системи спробують порівняти текстовий рядок `'dpt'` із числом 0, що є несумісною операцією, тому, зокрема, мій компілятор видає наступну помилку:

```
Conversion failed when converting the varchar value 'dpt' to data type int.
```

Питання 3. У цьому запиті, помилки не буде, і SQL системи переведуть текстовий рядок `'10000'` у число, щоб порівняти його з числом `salary`. Таким чином, результатом буде:

f_name	sex	age
Oleg	m	22
Anna	m	26
Galina	f	18

Питання 4. Команда `CAST` переведе текстовий рядок `'20160911'` у тип `DATE`. Результатом буде дата 2016-09-11, яка і виведеться на екран.

Питання 5. Команда спробує конкатенувати значення `age` та `'7000'` (наприклад, для співробітника Mashko Oleg, це буде 227000), а потім порівняти результат з `salary`. В цьому випадку, конкатенація дасть у вихід строку з числом, тому кастінг спрацює — програма не видасть помилку. Проте, через таку конкатенацію, число стає дуже великим і тому результатом буде пуста таблиця.

Питання 6. Тут все залежить від SQL системи. В двох різних компіляторах (MYSQL), в моєму випадку, видається два різних результати:

- 'salary' інтерпретується як текстовий рядок, тому порівняння з нулем дасть помилку: Conversion failed when converting the varchar value 'salary' to data type int.
- 'salary' інтерпретується як назва стовпця, тому порівняння з нулем буде виконано, але результатом буде уся таблиця.

Перший варіант виглядає більш логічно, а другий можливо спрацьовує через особливості реалізації SQL системи (котра, наприклад, прибирає цю помилку і виводить warning).

Питання 7. Маємо наступний запит:

```
1 SELECT CONCAT(l_name, ' ', f_name) AS full_name, age, sex, status,  
   ↪ position  
2 FROM employee  
3 ORDER BY age ASC;
```


4 Домашня Робота 4: Предикати

Умова Задачі 4.1. Таблиця та сама, що і в попередніх завданнях. За допомогою відомих предикатів:

1. Написати запит, щоб знайти всі зарплати між середньою та максимальною
2. Написати запит, щоб знайти всі зарплати між мінімальною та середньою за департаментами
3. Написати запит пошуку співробітника з ім'ям Oleg або Zahar
4. Написати запит, що виводить список співробітників віком від 22 до 26 років.
5. Порахувати кількість співробітників, які мають зарплату між 10000 і 20000
6. Порахувати кількість неодружених працівників від мінімального до середнього віку
7. Підрахувати кількість одружених співробітників на прізвище Petrenko

Відповідь.

Питання 1. Маємо наступний запит:

```
1 SELECT salary
2 FROM employee
3 WHERE salary BETWEEN (SELECT AVG(salary) FROM employee) AND (SELECT
  ↳ MAX(salary) FROM employee);
```

Тут, ми використали предикат BETWEEN, щоб вибрати всі зарплати, які знаходяться між середньою та максимальною зарплатою. Результатом будуть значення 20000, 25000, 18000.

Питання 2. Маємо наступний запит:

```
1 SELECT dpt, salary
2 FROM employee e
3 WHERE salary BETWEEN
4     (SELECT MIN(salary) FROM employee WHERE dpt = e.dpt) AND
5     (SELECT AVG(salary) FROM employee WHERE dpt = e.dpt);
```

Результатом буде наступна таблиця:

dpt	salary
IT	10000
IT	20000
Sales	8000

Питання 3. Маємо наступний запит:

```
1 SELECT *
2 FROM employee
3 WHERE f_name = 'Oleg' OR f_name = 'Zahar';
```

Питання 4. Маємо запит:

```
1 SELECT *
2 FROM employee
3 WHERE age BETWEEN 22 AND 26;
```

Питання 5. Використовуємо запит:

```
1 SELECT COUNT(*) AS employee_count
2 FROM employee
3 WHERE salary BETWEEN 10000 AND 20000;
```

Питання 6. Тут команда дещо складніша:

```
1 SELECT COUNT(*) AS single_employee_count
2 FROM employee
3 WHERE status = 'single'
4     AND age BETWEEN
5         (SELECT MIN(age) FROM employee)
6         AND (SELECT AVG(age) FROM employee);
```

Результатом буде просто 1. Якщо замінити COUNT(*) на *, то дізнаємось, що ця одна людина це Mashko Oleg.

Питання 7. Тут команда дещо простіша:

```
1 SELECT COUNT(*) AS married_petrenko_count
2 FROM employee
3 WHERE l_name = 'Petrenko' AND status = 'married';
```

5 Домашня Робота 5: Поєднання таблиць

Умова Задачі 5.1. Окрім таблиці employee, є ще таблиця departments:

dpt	nmb	rum_nmb	tel
IT	3	401	3444191
QA	2	402	3222311
Sales	2	403	3222233
PM	0	404	3224212
Design	0	405	3224444
Accounting	0	406	3221124

Завдання — написати запити (як з JOIN, так і без):

1. Вивести прізвища, імена та номери кімнат усіх співробітників.
2. Вивести списки працівників усіх непустих кімнат по номерам кімнат.
3. Вивести прізвища, імена, посади та номери кімнат всіх, хто працює у відділі Sales.
4. Вивести список кімнат (у тому числі і порожніх) з працюючими в них співробітниками, їх посадами та телефонами
5. Вивести прізвища, імена, вік, стать та зарплати співробітників, яких можна знайти за телефоном 3444191
6. Вивести найбільш повну інформацію про співробітників на ім'я Ivan.

Відповідь. Спочатку створимо таблицю departments:

```
1  -- Create the departments table
2  CREATE TABLE departments (
3      dpt VARCHAR(50),
4      nmb INT,
5      rum_nmb INT,
6      tel BIGINT
7  );
8
9  -- Insert data into the departments table
10 INSERT INTO departments (dpt, nmb, rum_nmb, tel) VALUES
11 ('IT', 3, 401, 3444191),
12 ('QA', 2, 402, 3222311),
13 ('Sales', 2, 403, 3222233),
14 ('PM', 0, 404, 3224212),
15 ('Design', 0, 405, 3224444),
16 ('Accounting', 0, 406, 3221124);
```

Питання 1. Маємо наступний запит:

```
1  SELECT e.l_name, e.f_name, d.rum_nmb
2  FROM employee e
3  JOIN departments d ON e.dpt = d.dpt;
```

Не використовуючи JOIN:

```
1 SELECT l_name, f_name, (SELECT rum_nmb FROM departments WHERE  
   ↪ dpt = employee.dpt) AS rum_nmb  
2 FROM employee;
```

Питання 2. Перший варіант наводимо без команди JOIN:

```
1 SELECT e.l_name, e.f_name,  
2 (SELECT rum_nmb FROM departments WHERE dpt = e.dpt) AS rum_nmb  
3 FROM employee e  
4 WHERE (SELECT nmb FROM departments WHERE dpt = e.dpt) > 0  
5 ORDER BY rum_nmb;
```

Далі, використовуючи JOIN:

```
1 SELECT d.rum_nmb, e.l_name, e.f_name  
2 FROM employee e  
3 JOIN departments d ON e.dpt = d.dpt  
4 WHERE d.nmb > 0  
5 ORDER BY d.rum_nmb;
```

Питання 3. Використовуємо JOIN:

```
1 SELECT e.l_name, e.f_name, e.position, d.rum_nmb  
2 FROM employee e  
3 JOIN departments d ON e.dpt = d.dpt  
4 WHERE d.dpt = 'Sales';
```

Команда, де ми не використовуємо JOIN:

```
1 SELECT e.l_name, e.f_name, e.position,  
2 (SELECT rum_nmb FROM departments WHERE dpt = e.dpt) AS rum_nmb  
3 FROM employee e  
4 WHERE e.dpt = 'Sales';
```

Питання 4. З командою JOIN:

```
1 SELECT d.rum_nmb, e.position, e.tel  
2 FROM departments d  
3 LEFT JOIN employee e ON d.dpt = e.dpt  
4 ORDER BY d.rum_nmb;
```

Без команди JOIN:

```
1 SELECT d.rum_nmb,  
2 (SELECT position FROM employee WHERE d.dpt = employee.dpt LIMIT  
   ↪ 1) AS position,  
3 (SELECT tel FROM employee WHERE d.dpt = employee.dpt LIMIT 1) AS
```

```
↪ tel
4 FROM departments d
5 ORDER BY d.rum_nmb;
```

Зауваження: тут ми використовуємо LIMIT 1, щоб вибрати лише одного співробітника з відділу, якщо їх кілька.

Питання 5. Якщо використовувати JOIN:

```
1 SELECT e.l_name, e.f_name, e.age, e.sex, e.salary
2 FROM employee e
3 JOIN departments d ON e.dpt = d.dpt
4 WHERE d.tel = 3444191;
```

Якщо не використовувати, то маємо

```
1 SELECT e.l_name, e.f_name, e.age, e.sex, e.salary
2 FROM employee e
3 WHERE (SELECT tel FROM departments WHERE dpt = e.dpt) = 3444191;
```

Питання 6. Використовуючи JOIN:

```
1 SELECT e.l_name, e.f_name, e.age, e.sex, e.status, e.position,
↪ e.salary, d.rum_nmb, d.tel
2 FROM employee e
3 JOIN departments d ON e.dpt = d.dpt
4 WHERE e.f_name = 'Ivan';
```

Якщо не використовувати, то маємо

```
1 SELECT e.l_name, e.f_name, e.age, e.sex, e.status, e.position,
↪ e.salary,
2     (SELECT rum_nmb FROM departments WHERE dpt = e.dpt) AS rum_nmb,
3     (SELECT tel FROM departments WHERE dpt = e.dpt) AS tel
4 FROM employee e
5 WHERE e.f_name = 'Ivan';
```

6 Домашня Робота 6: Підзапити. Псевдоніми

Умова Задачі 6.1. Маємо дві таблиці з попереднього завдання — employee та departments. За допомогою псевдонімів навести приклади для:

1. Використання псевдоніму для однієї таблиці.
2. Використання псевдоніму для однієї колонки таблиці.
3. Використання псевдоніму для результату агрегатної функції.
4. Використання псевдоніму для результату підзапиту.
5. Використання псевдонімів для таблиць та колонок при операції JOIN для 2х різних таблиць.
6. Використання псевдонімів для поєднання таблиці з самою собою

Відповідь.

Питання 1. Наприклад, можна скоротити назву таблиці employees до e:

```
1 SELECT e.l_name, e.f_name, e.age
2 FROM employee AS e
3 WHERE e.age > 25;
```

Результатом буде таблиця

l_name	f_name	age
Vovk	Zahar	32
Kononov	Oleg	26
Galenda	Anna	26

Питання 2. Можна, наприклад, змінити назву колонок l_name, f_name та salary:

```
1 SELECT l_name AS LastName, f_name AS FirstName, salary AS
   ↳ MonthlySalary
2 FROM employee
3 WHERE salary > 15000;
```

Результатом буде таблиця

LastName	FirstName	MonthlySalary
Mashko	Oleg	20000
Galenda	Anna	25000
Osokor	Galina	18000

Питання 3. Наприклад, можна вивести середню, максимальну та мінімальну зарплату:

```
1 SELECT AVG(salary) AS AverageSalary, MAX(salary) AS MaxSalary,
   ↳ MIN(salary) AS MinSalary
```

```
2 FROM employee;
```

Результатом буде проста таблиця, де AverageSalary=14916.66667, MaxSalary=25000, MinSalary=8000.

Питання 4. Обчислимо середню зарплатню і виведемо список співробітників із зарплатою вище середньої:

```
1 SELECT l_name AS LastName, f_name AS FirstName, salary AS Salary,
2        (SELECT AVG(salary) FROM employee) AS AverageSalary
3 FROM employee
4 WHERE salary > (SELECT AVG(salary) FROM employee);
```

Результатом буде таблиця

LastName	FirstName	Salary	AverageSalary
Mashko	Oleg	20000	14916
Galenda	Anna	25000	14916
Osokor	Galina	18000	14916

Питання 5. Виведемо прізвища, імена, зарплати співробітників, номер їх кімнат та телефони, використовуючи псевдоніми для таблиць та колонок:

```
1 SELECT
2     e.l_name AS LastName,
3     e.f_name AS FirstName,
4     e.salary AS Salary,
5     d.rum_nmb AS RoomNumber,
6     d.tel AS Phone
7 FROM employee AS e
8 JOIN departments AS d ON e.dpt = d.dpt;
```

Результатом буде таблиця

LastName	FirstName	Salary	RoomNumber	Phone
Osokor	Galina	18000	401	3444191
Galenda	Anna	25000	401	3444191
Petrenko	Zoya	10000	401	3444191
Mashko	Oleg	20000	401	3444191
Kononov	Oleg	8500	403	3222233
Vovk	Zahar	8000	403	3222233

Питання 6. Візьмемо такий приклад: знайти пари співробітників, де один заробляє більше іншого, і вивести лише їхні повні імена.

```
1 SELECT
2     CONCAT(e1.f_name, ' ', e1.l_name) AS Employee1_FullName,
3     CONCAT(e2.f_name, ' ', e2.l_name) AS Employee2_FullName
4 FROM employee AS e1
```

```
5 JOIN employee AS e2 ON e1.salary > e2.salary;
```

На виході маємо наступну таблицю:

Employee1_FullName	Employee2_FullName
Anna Galenda	Oleg Mashko
Galina Osokor	Zoya Petrenko
Anna Galenda	Zoya Petrenko
Oleg Mashko	Zoya Petrenko
Galina Osokor	Zahar Vovk
Anna Galenda	Zahar Vovk
Oleg Kononov	Zahar Vovk
Zoya Petrenko	Zahar Vovk
Oleg Mashko	Zahar Vovk
Galina Osokor	Oleg Kononov
Anna Galenda	Oleg Kononov
Zoya Petrenko	Oleg Kononov
Oleg Mashko	Oleg Kononov
Anna Galenda	Galina Osokor
Oleg Mashko	Galina Osokor

7 Домашня Робота 7: Регулярні вирази

Умова Задачі 7.1. Є таблиця employee з попередніх завдань. За допомогою регулярних виразів:

1. Вибрати прізвища та посади всіх, у кого у прізвищі, або в імені є nko.
2. Вибрати імена, прізвища, посади співробітників, які мають у віці цифру 4 або 6.
3. Вибрати прізвища та посади всіх співробітників із зарплатами рівно з трьома 0.
4. Вивести список усіх співробітників з ім'ям або прізвищем на літеру O
5. Вивести всі дані людини, у якої в даних статусу є літери g, e, але немає літери l
6. Що виведе наступний запит

```
1 SELECT fname, lname, age FROM employee WHERE fname LIKE '%k%'
```

7. Написати регулярний вираз для пошуку подвоєних приголосних у будь-якій кількості повторень (без прив'язки до таблиці)
8. Написати регулярний вираз для пошуку % в полі position, % - це звичайний символ (використати escape конструкцію).

Відповідь.

Питання 1. Маємо наступний запит:

```
1 SELECT l_name AS LastName, position AS Position
2 FROM employee
3 WHERE l_name LIKE '%nko%' OR f_name LIKE '%nko%';
```

Виведе лише один рядок: Petrenko, Programmer.

Питання 2. Маємо наступний запит:

```
1 SELECT f_name AS FirstName, l_name AS LastName, position AS Position
2 FROM employee
3 WHERE CAST(age AS CHAR) LIKE '%4%' OR CAST(age AS CHAR) LIKE '%6%';
```

На виході отримаємо три рядки:

FirstName	LastName	Position
Zoya	Petrenko	Programmer
Anna	Galenda	Programmer
Oleg	Kononov	Sales

Питання 3. Маємо наступний запит:

```
1 SELECT l_name AS LastName, position AS Position
2 FROM employee
3 WHERE CAST(salary AS CHAR) LIKE '%000';
```

На виході отримаємо п'ять рядків:

LastName	Position
Mashko	Admin
Petrenko	Programmer
Vovk	Admin
Galenda	Programmer
Osokor	QA

Питання 4. Маємо такий запит:

```
1 SELECT f_name AS FirstName, l_name AS LastName, position AS Position
2 FROM employee
3 WHERE f_name LIKE 'O%' OR l_name LIKE 'O%';
```

На виході отримуємо три рядки:

FirstName	LastName	Position
Oleg	Mashko	Admin
Oleg	Kononov	Sales
Galina	Osokor	QA

Питання 5. Для цього пункту маємо наступний запит:

```
1 SELECT *
2 FROM employee
3 WHERE status LIKE '%g%'
4     AND status LIKE '%e%'
5     AND status NOT LIKE '%l%';
```

Питання 6. Потрібно проаналізувати запит

```
1 SELECT fname, lname, age
2 FROM employee
3 WHERE fname LIKE '%k%'
```

Ця умова шукає всі записи, у яких ім'я (f_name) містить літеру k у будь-якому місці (на початку, всередині або в кінці). Символи % означають, що перед і після літери k можуть бути будь-які символи. На виході буде пуста таблиця, бо в таблиці немає жодного співробітника з ім'ям, яке містить літеру k.

Питання 7. Маємо такий вираз: $([bcdfghjklmnpqrstvwxyz]\{2,\})^+$. По порядку:

- $[bcdfghjklmnpqrstvwxyz]$ — всі малі приголосні латинського алфавіту. Можна врахувати і великі, тоді буде $[bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ]$.
- $\{2,\}$: Це вказує, що потрібно знайти дві або більше однакових приголосних, що йдуть підряд.
- $^+$: Квантифікатор + дозволяє знаходити повторення таких послідовностей приголосних. Тобто він шукає кілька таких подвоєних послідовностей приголосних.

Питання 8. В довільній таблиці можна використати вираз:

```
1 SELECT *  
2 FROM your_table  
3 WHERE position REGEXP '\\%';
```

В SQL для правильного використання зворотного слешу в регулярному виразі потрібно подвоїти зворотний слеш, тому що один зворотний слеш буде екрануватися.

8 Домашня Робота 8: Створення, редагування, видалення

Умова Задачі 8.1.

1. Створіть бази даних test1 і test2.
2. Видаліть базу даних test2.
3. Створіть у базі даних test1 таблицю table_1 (з полями id — int, firstname — varchar, lastname — varchar, position — varchar, department — varchar, salary — decimal, age — int, birthdate — date) та таблицю table_2 (з полями department — varchar, qtyemployees — int, head — varchar).
4. Перейменуйте таблицю table_1 на employees.
5. Видаліть таблицю table_2.
6. Додайте в таблицю employees поле status — varchar(255).
7. Видаліть із таблиці employees поле age.
8. Перейменуйте в employees поле firstname на name.
9. Змініть тип поля age з int на varchar(50).
10. Додайте кілька записів, а потім очистіть таблицю employee.
11. Очистіть усі таблиці бази даних test_1, потім видаліть її.

Відповідь.

Питання 1. Створюємо бази даних

```
1 CREATE DATABASE test1;  
2 CREATE DATABASE test2;
```

Питання 2. Видаляємо базу даних test2

```
1 DROP DATABASE test2;
```

Питання 3. Створюємо таблиці table_1 та table_2 у базі даних test1:

```
1 USE test1;  
2  
3 -- Creating table_1  
4 CREATE TABLE table_1 (  
5     id INT,  
6     firstname VARCHAR(255),  
7     lastname VARCHAR(255),  
8     position VARCHAR(255),  
9     department VARCHAR(255),  
10    salary DECIMAL(10, 2),  
11    age INT,  
12    birthdate DATE  
13 );  
14  
15 -- Creating table_2
```

```
16 CREATE TABLE table_2 (  
17     department VARCHAR(255),  
18     qtyemployees INT,  
19     head VARCHAR(255)  
20 );
```

Питання 4. Перейменовуємо таблицю table_1 на employees:

```
1 RENAME TABLE table_1 TO employees;
```

Питання 5. Видаляємо таблицю table_2:

```
1 DROP TABLE table_2;
```

Питання 6. Додаємо поле status у таблицю employees:

```
1 ALTER TABLE employees  
2 ADD COLUMN status VARCHAR(255);
```

Питання 7. Видаляємо поле age з таблиці employees:

```
1 ALTER TABLE employees  
2 DROP COLUMN age;
```

Питання 8. Перейменовуємо поле firstname на name:

```
1 ALTER TABLE employees  
2 CHANGE COLUMN firstname name VARCHAR(255);
```

Питання 9. Оскільки поле age вже видалено, то наведемо приклад для зміни типу поля salary з int на VARCHAR(50):

```
1 ALTER TABLE employees  
2 MODIFY COLUMN salary VARCHAR(50);
```

Питання 10. Додаємо кілька записів у таблицю employees:

```
1 -- Додавання кількох записів  
2 INSERT INTO employees (id, name, lastname, position, department,  
3     ↳ salary, birthdate, status)  
4 VALUES  
5 (1, 'Dmytro', 'Zakharov', 'Student', 'Sales', 5000.00, '1985-06-15',  
6     ↳ 'Active'),  
7 (2, 'Petya', 'Ugulkin', 'Developer', 'IT', 4500.00, '1990-02-20',  
8     ↳ 'Active'),  
9 (3, 'Vasya', 'Pestolkin', 'Designer', 'Design', 4000.00,  
10    ↳ '1988-11-10', 'Inactive');  
11  
12 -- Clearing table employees  
13 TRUNCATE TABLE employees;
```

Питання 11. Очищення усіх таблиць бази даних test1 та видалення її:

```
1 USE test1;  
2  
3 -- Clearing all tables  
4 TRUNCATE TABLE employees;  
5  
6 -- Dropping the database test1  
7 DROP DATABASE test1;
```