Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №3

по дисциплине «Программно-аппаратные средства хранения и обработки данных»

на тему «Создание простых и сложных запросов в среде MySQL»

Выполнили:

студенты группы 22ВВП1

Хоссейни Нежад С.А.С.М.

Захаров А. С.

Сергунов М. Р.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2025

**Название**

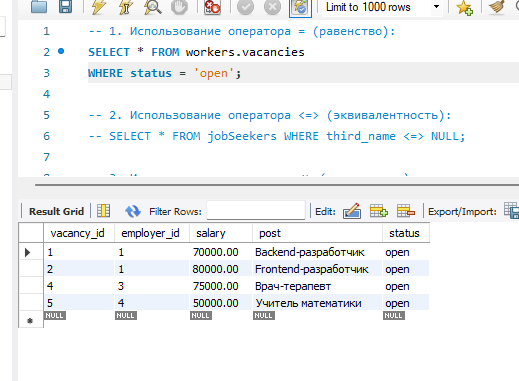
Создание простых и сложных запросов в среде MySQL

**Цель работы**

Научиться создавать простые и сложные запросы в среде MySQL.

**Ход работы:**

1. **Создали 12 простых запросов с использованием операторов и встроенных функций в MySQL.**



**Листинг:**

-- 1. Использование оператора = (равенство):

SELECT \* FROM workers.vacancies

WHERE status = 'open';

-- 2. Использование оператора <=> (эквивалентность):

-- SELECT \* FROM jobSeekers WHERE third\_name <=> NULL;

-- 3. Использование оператора <> (неравенство):

-- SELECT \* FROM employers WHERE activity\_id <> 2;

-- 4. Использование оператора < (меньше):

-- SELECT \* FROM vacancies WHERE salary < 70000;vacancies

-- 5. Использование оператора <= (меньше или равно):

-- SELECT \* FROM vacancies WHERE salary <= 75000;

-- 6. Использование оператора > (больше):

-- SELECT \* FROM vacancies WHERE salary > 50000;

-- 7. Использование оператора >= (больше или равно):

-- SELECT \* FROM vacancies WHERE salary >= 80000;

-- 8. Использование оператора BETWEEN ... AND ... (проверка диапазона):

-- SELECT \* FROM deals WHERE comission BETWEEN 4000 AND 5000;

-- 9. Использование операторов IS NULL и IS NOT NULL:

-- SELECT \* FROM jobSeekers WHERE third\_name IS NULL;

-- SELECT \* FROM jobSeekers WHERE third\_name IS NOT NULL;

-- 10. Использование оператора IN (принадлежность множеству):

-- SELECT \* FROM employers WHERE activity\_id IN (1, 3);

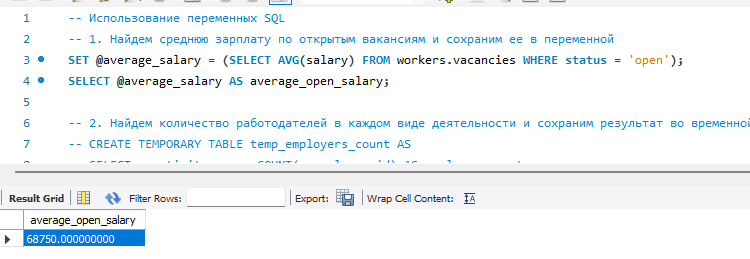
-- 11. Использование оператора AND (логическое И):

-- SELECT \* FROM vacancies WHERE salary > 50000 AND status = 'open';

-- 12. Использование оператора OR (логическое ИЛИ):

-- SELECT \* FROM vacancies WHERE salary < 60000 OR status = 'closed';

1. **Создали 4 запроса с использованием переменных MySQL и временные таблицы**



**Листинг:**

-- Использование переменных SQL

-- 1. Найдем среднюю зарплату по открытым вакансиям и сохраним ее в переменной

SET @average\_salary = (SELECT AVG(salary) FROM workers.vacancies WHERE status = 'open');

SELECT @average\_salary AS average\_open\_salary;

-- 2. Найдем количество работодателей в каждом виде деятельности и сохраним результат во временной таблице

-- CREATE TEMPORARY TABLE temp\_employers\_count AS

-- SELECT a.activity\_name, COUNT(e.employer\_id) AS employer\_count

-- FROM employers e

-- JOIN activityTypes a ON e.activity\_id = a.activity\_id

-- GROUP BY a.activity\_name;

-- Проверим содержимое временной таблицы

-- SELECT \* FROM temp\_employers\_count;

-- 3. Найдем общее количество сделок и максимальную комиссию по ним, сохранив их в переменные

-- SELECT COUNT(\*) INTO @total\_deals FROM deals;

-- SELECT MAX(comission) INTO @max\_comission FROM deals;

-- SELECT @total\_deals AS total\_deals, @max\_comission AS max\_comission;

-- 4. Создадим временную таблицу с данными о вакансиях и соответствующими работодателями

-- CREATE TEMPORARY TABLE temp\_vacancies AS

-- SELECT v.vacancy\_id, e.company\_name, v.salary, v.status

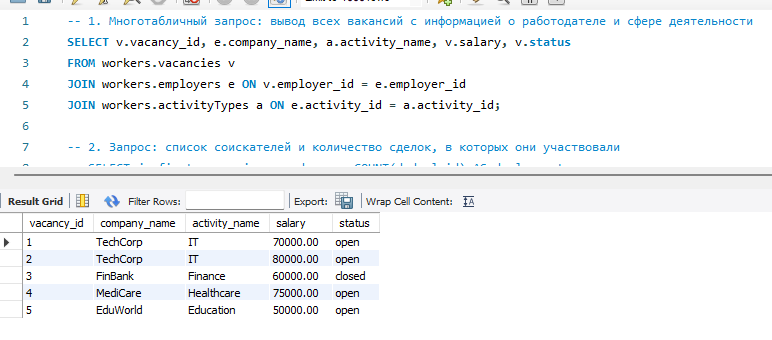
-- FROM vacancies v

-- JOIN employers e ON v.employer\_id = e.employer\_id;

-- Проверим содержимое временной таблицы

-- SELECT \* FROM temp\_vacancies;

1. **Создание 10 сложных запросов к таблице**



**Листинг:**

-- 1. Многотабличный запрос: вывод всех вакансий с информацией о работодателе и сфере деятельности

SELECT v.vacancy\_id, e.company\_name, a.activity\_name, v.salary, v.status

FROM workers.vacancies v

JOIN workers.employers e ON v.employer\_id = e.employer\_id

JOIN workers.activityTypes a ON e.activity\_id = a.activity\_id;

-- 2. Запрос: список соискателей и количество сделок, в которых они участвовали

-- SELECT js.first\_name, js.second\_name, COUNT(d.deal\_id) AS deal\_count

-- FROM jobSeekers js

-- LEFT JOIN deals d ON js.seekers\_id = d.seeker\_id

-- GROUP BY js.first\_name, js.second\_name;

--

-- -- 3. Запрос: работодатели, у которых есть открытые вакансии

-- SELECT e.company\_name, COUNT(v.vacancy\_id) AS open\_vacancies

-- FROM employers e

-- JOIN vacancies v ON e.employer\_id = v.employer\_id

-- WHERE v.status = 'open'

-- GROUP BY e.company\_name;

--

-- -- 4. Вложенный запрос: найти работодателей, у которых есть вакансии с зарплатой выше средней

-- SELECT company\_name FROM employers

-- WHERE employer\_id IN (

-- SELECT employer\_id FROM vacancies

-- WHERE salary > (SELECT AVG(salary) FROM vacancies)

-- );

--

-- -- 5. Запрос: список вакансий с самыми высокими зарплатами в каждой сфере деятельности

-- SELECT v.vacancy\_id, e.company\_name, a.activity\_name, v.salary

-- FROM vacancies v

-- JOIN employers e ON v.employer\_id = e.employer\_id

-- JOIN activityTypes a ON e.activity\_id = a.activity\_id

-- WHERE v.salary = (

-- SELECT MAX(salary) FROM vacancies v2

-- JOIN employers e2 ON v2.employer\_id = e2.employer\_id

-- WHERE e2.activity\_id = a.activity\_id

-- );

--

-- -- 6. Запрос: работодатели, у которых нет активных вакансий

-- SELECT e.company\_name FROM employers e

-- WHERE NOT EXISTS (

-- SELECT 1 FROM vacancies v

-- WHERE v.employer\_id = e.employer\_id AND v.status = 'open'

-- );

--

-- -- 7. Запрос: список вакансий, на которые не было заключено сделок

-- SELECT v.vacancy\_id, e.company\_name, v.salary

-- FROM vacancies v

-- JOIN employers e ON v.employer\_id = e.employer\_id

-- WHERE v.vacancy\_id NOT IN (SELECT vacancy\_id FROM deals);

--

-- -- 8. Запрос: средняя комиссия за сделки в каждой сфере деятельности

-- SELECT a.activity\_name, AVG(d.comission) AS avg\_comission

-- FROM deals d

-- JOIN employers e ON d.employer\_id = e.employer\_id

-- JOIN activityTypes a ON e.activity\_id = a.activity\_id

-- GROUP BY a.activity\_name;

--

-- -- 9. Запрос: список соискателей, у которых нет заключенных сделок

-- SELECT js.first\_name, js.second\_name FROM jobSeekers js

-- WHERE NOT EXISTS (

-- SELECT 1 FROM deals d WHERE d.seeker\_id = js.seekers\_id

-- );

--

-- -- 10. Запрос: вакансии с максимальной зарплатой среди открытых вакансий

-- SELECT v.vacancy\_id, e.company\_name, v.salary FROM vacancies v

-- JOIN employers e ON v.employer\_id = e.employer\_id

-- WHERE v.salary = (SELECT MAX(salary) FROM vacancies WHERE status = 'open');

**Вывод:**

Мы научились создавать простые запросы к таблице с использованием операторов и встроенных функций в MySQL, переменных и встроенных таблиц, а также делать сложные запросы к таблице.