|  |
| --- |
| Объектно-реляционные базы данных. Манипуляция данными и пользовательские операторы |

1. **Вариант задания (12 вариант)**

Вакансии: волонтерские позиции, название вакансии, организация работодатель, адрес работодателя, диапазон зарплаты, требования к образованию, Обязанности, график работы, требования обязательные, желательные, дата выставления вакансии.

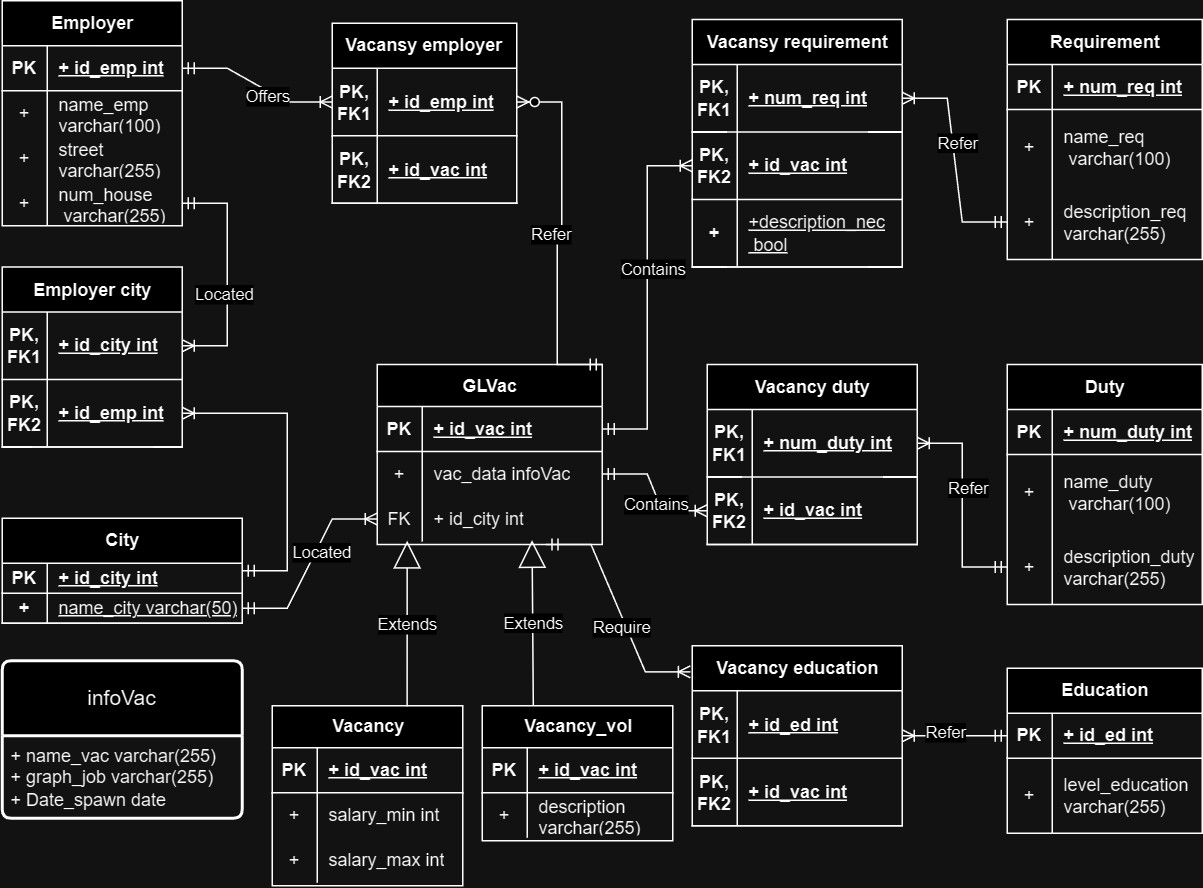
а. вакансии, имеющие в названии SQL, но не заканчивающиеся на него

б. работодатели в Санкт-Петербурге, выставившие несколько вакансий

в. вакансия с наибольшей зарплатой

г. волонтерские позиции с максимальным количеством требований

д. вакансии, в которых нет требования к опыту работы

2. **Физическая модель базы данных:** 

1. **Выполнение заданий**

**Вставка данных:**

INSERT INTO City (name\_city) VALUES ('Санкт-Петербург');

INSERT INTO City (name\_city) VALUES ('Москва');

INSERT INTO Vacancy (vac\_data, salary\_min, salary\_max, id\_city) VALUES (ROW('frontend junior', '2/2', '2024-04-10'),40000, 50000,1);

INSERT INTO Vacancy (vac\_data, salary\_min, salary\_max, id\_city) VALUES (ROW('backend junior', '5/2', '2024-04-11'),50000, 60000,2);

INSERT INTO Vacancy\_vol (vac\_data, description,id\_city) VALUES (ROW('Волонтер на хоккейную арену', '2/2', '2024-04-13'),'Есть возможность бесплатно смотреть матчи', 2);

INSERT INTO Vacancy\_vol (vac\_data, description,id\_city) VALUES (ROW('Волонтер на футбольный матч', '5/2', '2024-04-12'),'Обеды обеспечены', 2);

INSERT INTO Employer (name\_emp, street, num\_house) VALUES ('Pocket', 'Казанская', '5');

INSERT INTO Employer (name\_emp, street, num\_house) VALUES ('Omega', 'Ленина', '7');

INSERT INTO Employer\_city VALUES (1, 1);

INSERT INTO Employer\_city VALUES (2, 2);

INSERT INTO Employer\_city VALUES (2, 3);

INSERT INTO Vacancy\_employer VALUES (1, 1);

INSERT INTO Vacancy\_employer VALUES (1, 2);

INSERT INTO Vacancy\_employer VALUES (2, 3);

INSERT INTO Vacancy\_employer VALUES (2, 4);

INSERT INTO Requirement (name\_req, description\_req) VALUES ('Опрятность', 'Необходимо иметь опрятный вид');

INSERT INTO Requirement (name\_req, description\_req) VALUES ('Пунктуальность', 'Необходимо всегда приходить вовремя');

INSERT INTO Requirement (name\_req, description\_req) VALUES ('Стаж 2г', 'Необходимо иметь стаж в этой области более двух лет');

INSERT INTO Requirement (name\_req, description\_req) VALUES ('Хорошая дикция', 'Необходимо обладать хорошей дикцией');

INSERT INTO Vacancy\_requirement VALUES (1, 1, false);

INSERT INTO Vacancy\_requirement VALUES (2, 1, true);

INSERT INTO Vacancy\_requirement VALUES (2, 2, false);

INSERT INTO Vacancy\_requirement VALUES (3, 2, false);

INSERT INTO Vacancy\_requirement VALUES (1, 3, false);

INSERT INTO Vacancy\_requirement VALUES (4, 4, false);

INSERT INTO Duty (name\_duty, description\_duty) VALUES ('Обеспечение it жизнедеятельности', 'Обеспечение стабильного и безотказного функционирования ИТ-инфраструктуры');

INSERT INTO Duty (name\_duty, description\_duty) VALUES ('Водить автомобиль', 'Управлять автомобилем категории B');

INSERT INTO Duty (name\_duty, description\_duty) VALUES ('Помощь людям', 'Необходимо отвечать на вопросы и всячески помогать людям');

INSERT INTO Vacancy\_duty VALUES (1, 1);

INSERT INTO Vacancy\_duty VALUES (1, 2);

INSERT INTO Vacancy\_duty VALUES (2, 3);

INSERT INTO Vacancy\_duty VALUES (3, 2);

INSERT INTO Education (level\_education) VALUES ('Без образования');

INSERT INTO Education (level\_education) VALUES ('Основное общее образование');

INSERT INTO Vacancy\_education VALUES (2, 1);

INSERT INTO Vacancy\_education VALUES (2, 2);

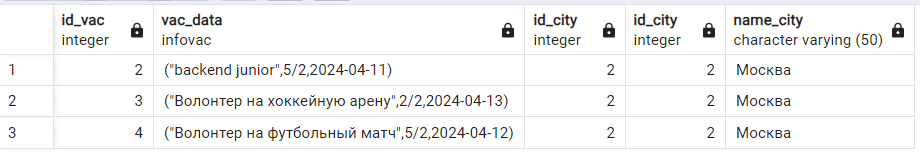
INSERT INTO Vacancy\_education VALUES (1, 3);

INSERT INTO Vacancy\_education VALUES (1, 4);

**Запрос выборки с условием к таблицам и предку и потомку:**

SELECT \* FROM GLVac left join City on GLVac.id\_city = City.id\_city

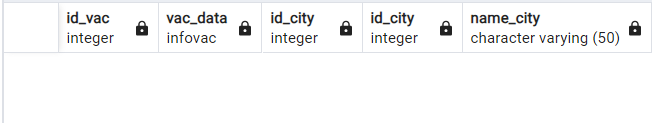
WHERE City.name\_city = 'Москва';



**Запрос выборки с условием только к таблице предку:**

SELECT \* FROM only GLVac left join City on GLVac.id\_city = City.id\_city

WHERE City.name\_city = 'Москва';



**Запрос выборки с условием только к таблице потомку:**

SELECT \* FROM Vacancy left join City on Vacancy.id\_city = City.id\_city

WHERE City.name\_city = 'Москва'; 

**Пользовательский оператор:**

CREATE OR REPLACE FUNCTION vac\_graph\_eq(infoVac, infoVac)

RETURNS BOOLEAN LANGUAGE SQL IMMUTABLE

AS $$

SELECT $1.graph\_job = $2.graph\_job;

$$;

CREATE OPERATOR %= (

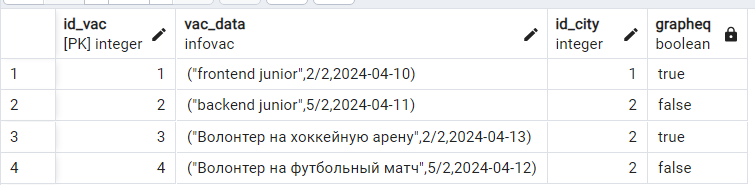
PROCEDURE = vac\_graph\_eq,

LEFTARG = infoVac,

RIGHTARG = infoVac

);

SELECT \*,vac.vac\_data %= (SELECT vac\_data FROM GLVac AS another WHERE another.id\_vac = 1) as graphEq FROM GLVac AS vac



**Агрегатная функция:**

CREATE OR REPLACE FUNCTION data\_max(ac1 infoVac ,ac2 infoVac )

RETURNS infoVac LANGUAGE plpgsql

AS

$$begin

IF ac1.date\_spawn IS NULL

THEN

RETURN ac2;

ELSEIF ac2.date\_spawn IS NULL

THEN RETURN ac1;

ELSEIF ac1.date\_spawn>=ac2.date\_spawn

THEN RETURN ac1;

else RETURN ac2;

END IF;

end; $$

IMMUTABLE RETURNS NULL ON NULL INPUT;

CREATE AGGREGATE max (infoVac)

(

sfunc = data\_max,

stype = infoVac

);

SELECT max(GLVac.vac\_data) from GLVac group by id\_vac;

