Laboratory work 2 Variant 2 Level 1

1. Creating tables

```
CREATE TYPE MONTH AS ENUM ('Январь', 'Февраль', 'Март', 'Апрель',
'Май', 'Июнь', 'Июль', 'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь');
CREATE TABLE Employers (
      id SERIAL PRIMARY KEY,
      name TEXT NOT NULL UNIQUE.
      location TEXT NOT NULL,
      benefit SERIAL
);
CREATE TABLE Recruitment_bureaus (
      id SERIAL PRIMARY KEY.
      no TEXT NOT NULL UNIQUE,
      address TEXT NOT NULL,
      payment_for_services SERIAL
);
CREATE TABLE Professions (
      id SERIAL PRIMARY KEY,
      name TEXT NOT NULL UNIQUE,
      cost of hiring SERIAL,
      number SERIAL,
      last_job_place TEXT
);
CREATE TABLE Employment_contracts (
      no SERIAL PRIMARY KEY,
      date MONTH NOT NULL,
      employer_id SERIAL NOT NULL REFERENCES Employers (id),
      recruitment_bureau_id SERIAL NOT NULL REFERENCES
      Recruitment bureaus (id),
      profession_id SERIAL NOT NULL REFERENCES Professions (id),
      vacancies number SERIAL,
      payment SERIAL
);
```

Пояснение использования типов:

- 1) Employers.id неотрицательное число, Primary key включает в себя UNIQUE и NOT NULL
- 2) Employers.name строка, имя не может быть нулевым и повторяющимся
- 3) Employers.location строка, расположении не может быть нулевым, но может быть повторяющимся, т.к. в данной реализации это только город.
- 4) Employers.benefit неотрицательное число, льгота может быть нулевой и может повторятся
- 5) Recruitment_bureaus.id неотрицательное число, Primary key включает в себя UNIQUE и NOT NULL
- 6) Recruitment_bureaus.no строка типа 'N%', номер не может быть нулевым или повторяющимся
- 7) Recruitment_bureaus.address строка, адрес не может быть нулевым, но может быть повторяющимся, т.к. в данной реализации это только город
- 8) Recruitment_bureaus payment_for_servieces неотрицательное число, оплата теоретически может равняться нулю и быть повторяющейся
- 9) Professions.id неотрицательное число, Primary key включает в себя UNIQUE и NOT NULL
- 10) Professions.name строка, название профессии не может быть нулевым и повторяющимся
- 11) Professions.cost_of_hiring неотрицательное число, стоимость найма теоретически может равняться нулю и быть повторяющейся
- 12) Professions.number неотрицательное число, количество может быть нулевым и повторяющимся
- 13) Employment_contracts.no неотрицательное число, Primary key включает в себя UNIQUE и NOT NULL
- 14) Employment_contracts.date ввел свой тип данных для месяцов, дата не может быть нулевой, но может быть повторяющейся
- 15) Employment contracts.employer id описано в п.1
- 16) Employment_contracts.recruitment_bureau_id описано в п. 5
- 17) Employment contracts.profession id описано в п. 9
- 18) Employment_contracts.vacancies_number неотрицательное число, может равняться нулю и повторяться
- 19) Employment_contracts.payment неотрицательное число, оплата теоретически может равняться нулю и быть повторяющейся

2. Filling tables

```
INSERT INTO Employers(id, name, location, benefit)
VALUES (1, 'Pora и копыта', 'Приморск', 0),
(2, 'ГАЗ', 'H. Новгород', 20),
(3, 'Станкостроительный завод', 'Одесса', 2),
(4, 'Кинап', 'Одесса', 2),
(5, 'КРАЗ', 'Кременчуг', 2),
(6, 'п/я 12687-у', 'Capaнск', 10);
INSERT INTO Recruitment_bureaus(id, no, address, payment_for_services)
```

VALUES (1, 'N5', 'H. Новгород', 4),

```
(2, 'N4', 'H. Москва', 3),
(3, 'N12', 'Киев', 11),
(4, 'N6', 'Н. Новгород', 3),
(5, 'N8', 'Одесса', 9);
INSERT INTO Professions(id, name, cost_of_hiring, "number", last_job_place)
VALUES (1, 'Кровельщик', 10000, 7, 'Саранск'),
(2, 'Слесарь', 15000, 6, 'Кременчуг'),
(3, 'Счетовод', 25000, 10, 'Москва'),
(4, 'Фрезеровщик', 20000, 7, 'Одесса'),
(5, 'Программист', 40000, 8, 'Киев'),
(6, 'Автоводитель', 25000, 3, 'Приморск'),
(7, 'Шлифовальщик', 17000, 5, 'Одесса');
INSERT INTO Employment_contracts(no, date, employer_id, recruitment_bureau_id,
profession_id, vacancies_number, payment)
VALUES (127, 'Январь', 3, 4, 6, 1, 25000),
(128, 'Февраль', 6, 2, 1, 2, 20000),
(129, 'Mapt', 1, 3, 4, 1, 20000),
(130, 'Апрель', 2, 1, 7, 2, 34000),
(131, 'Апрель', 4, 4, 6, 1, 25000),
(132, 'Апрель', 6, 4, 1, 1, 10000),
(133, 'Май', 5, 2, 5, 3, 120000),
(134, 'Май', 3, 3, 2, 3, 45000),
(135, 'Май', 3, 4, 1, 1, 10000),
(136, 'Июнь', 4, 1, 3, 4, 100000),
(137, 'Июнь', 1, 2, 2, 3, 45000),
(138, 'Июнь', 1, 3, 7, 1, 17000),
(139, 'Июнь', 5, 4, 1, 2, 20000),
(140, 'Июнь', 3, 5, 2, 1, 15000),
(141, 'Июнь', 3, 5, 2, 1, 15000),
(142, 'Июнь', 4, 2, 6, 1, 25000),
(143, 'Сентябрь', 2, 2, 7, 2, 34000);
```

3. The result of creating and filling

1) SELECT * FROM Employers

	id [PK] integer	name text	location text	benefit integer
1	1	Рога и копыта	Приморск	0
2	2	ГАЗ	Н. Новгород	20
3	3	Станкостроительный зав	Одесса	2
4	4	Кинап	Одесса	2
5	5	KPA3	Кременчуг	2
6	6	п/я 12687-у	Саранск	10

2) SELECT * FROM Recruitment_bureaus

	id [PK] integer	no text	address text	payment_for_services , integer
1	1	N5	Н. Новго	4
2	2	N4	Н. Москва	3
3	3	N12	Киев	11
4	4	N6	Н. Новго	3
5	5	N8	Одесса	9

3) SELECT * FROM Professions

	id [PK] integer	name text	cost_of_hiring /	number integer	last_job_place / text
1	1	Кровель	10000	7	Саранск
2	2	Слесарь	15000	6	Кременчуг
3	3	Счетовод	25000	10	Москва
4	4	Фрезеро	20000	7	Одесса
5	5	Програм	40000	8	Киев
6	6	Автовод	25000	3	Приморск
7	7	Шлифов	17000	5	Одесса

4) SELECT * FROM Employment_contracts

	no [PK] integer	date month	emloyer_id integer	recruitment_bureau_id /	profession_id integer	vacancies_number /	payment /
1	127	Январь	3	4	6	1	25000
2	128	Февраль	6	2	1	2	20000
3	129	Март	1	3	4	1	20000
4	130	Апрель	2	1	7	2	34000
5	131	Апрель	4	4	6	1	25000
6	132	Апрель	6	4	1	1	10000
7	133	Май	5	2	5	3	120000
8	134	Май	3	3	2	3	45000
9	135	Май	3	4	1	1	10000
10	136	Июнь	4	1	3	4	100000
11	137	Июнь	1	2	2	3	45000
12	138	Июнь	1	3	7	1	17000
13	139	Июнь	5	4	1	2	20000
14	140	Июнь	3	5	2	1	15000
15	141	Июнь	3	5	2	1	15000
16	142	Июнь	4	2	6	1	25000
17	143	Сентябрь	2	2	7	2	34000

4. SELECT

1) SELECT name, location FROM Employers

	name text	location text
1	Рога и ко	Приморск
2	ГАЗ	Н. Новго
3	Станкост	Одесса
4	Кинап	Одесса
5	KPA3	Кременчуг
6	п/я 1268	Саранск

2) SELECT no FROM Recruitment_bureaus

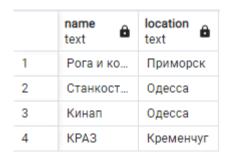
=+	• ~	
	no text	â
1	N5	
2	N4	
3	N12	
4	N6	
5	N8	

3) SELECT name, number FROM Professions

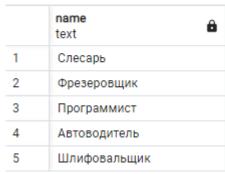
	name text	number integer
1	Кровель	7
2	Слесарь	6
3	Счетовод	10
4	Фрезеро	7
5	Програм	8
6	Автовод	3
7	Шлифов	5

5. SELECT + WHERE

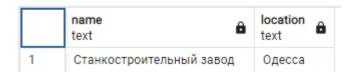
 SELECT name, location FROM Employers WHERE benefit < 8



2) SELECT name FROM Professions WHERE cost_of_hiring > 10000 AND last_job_place != 'Москва'



3) SELECT name, location FROM Employers WHERE name like '%завод%' AND benefit > 0



6. SELECT + JOIN

SELECT name, date, no FROM Employment_contracts
 JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id
 JOIN Employers ON Employment_contracts.employer_id = Employers.id;

Пояснение: в задании указано вывести название бюро, но такого столбца нет в таблице Recruitment_bureaus, вместо него был выведет номер (no).

	name text	date month	no text
1	Станкостроительный завод	Январь	N6
2	п/я 12687-у	Февраль	N4
3	Рога и копыта	Март	N12
4	ГАЗ	Апрель	N5
5	Кинап	Апрель	N6
6	п/я 12687-у	Апрель	N6
7	KPA3	Май	N4
8	Станкостроительный завод	Май	N12
9	Станкостроительный завод	Май	N6
10	Кинап	Июнь	N5
11	Рога и копыта	Июнь	N4
12	Рога и копыта	Июнь	N12
13	KPA3	Июнь	N6
14	Станкостроительный завод	Июнь	N8
15	Станкостроительный завод	Июнь	N8
16	Кинап	Июнь	N4
17	ГАЗ	Сентябрь	N4

2) SELECT date, Recruitment_bureaus.no, name, vacancies_number FROM Employment_contracts

JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id

JOIN Professions ON Employment_contracts.profession_id = Professions.id;

	date month	no text	name text	vacancies_number integer
1	Январь	N6	Автовод	1
2	Февраль	N4	Кровель	2
3	Март	N12	Фрезеро	1
4	Апрель	N5	Шлифов	2
5	Апрель	N6	Автовод	1
6	Апрель	N6	Кровель	1
7	Май	N4	Програм	3
8	Май	N12	Слесарь	3
9	Май	N6	Кровель	1
10	Июнь	N5	Счетовод	4
11	Июнь	N4	Слесарь	3
12	Июнь	N12	Шлифов	1
13	Июнь	N6	Кровель	2
14	Июнь	N8	Слесарь	1
15	Июнь	N8	Слесарь	1
16	Июнь	N4	Автовод	1
17	Сентябрь	N4	Шлифов	2

7. SELECT + JOIN + WHERE

 SELECT date, Employment_contracts.no, Employers.name FROM Employment_contracts

JOIN Professions ON Employment_contracts.profession_id = Professions.id JOIN Employers ON employment_contracts.employer_id = Employers.id WHERE Professions.name = 'Автоводитель' OR payment >= 14000;

	date month	no integer	name text
1	Январь	127	Станкостроительный зав
2	Февраль	128	п/я 12687-у
3	Март	129	Рога и копыта
4	Апрель	130	ГАЗ
5	Апрель	131	Кинап
6	Май	133	KPA3
7	Май	134	Станкостроительный зав
8	Июнь	136	Кинап
9	Июнь	137	Рога и копыта
10	Июнь	138	Рога и копыта
11	Июнь	139	KPA3
12	Июнь	140	Станкостроительный зав
13	Июнь	141	Станкостроительный зав
14	Июнь	142	Кинап
15	Сентябрь	143	ГАЗ

2) SELECT Recruitment_bureaus.no, address FROM Employment_contracts
JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id =
Recruitment_bureaus.id
JOIN Employers ON Employment_contracts.employer_id = Employers.id
WHERE benefit < 7 AND date > 'Февраль'

=+	□ ∨ □	
	no text	address text
1	N12	Киев
2	N6	Н. Новго
3	N4	Н. Москва
4	N12	Киев
5	N6	Н. Новго
6	N5	Н. Новго
7	N4	Н. Москва
8	N12	Киев
9	N6	Н. Новго
10	N8	Одесса
11	N8	Одесса
12	N4	Н. Москва

3) SELECT DISTINCT name FROM Employment_contracts JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id JOIN Employers ON Employment_contracts.employer_id = Employers.id WHERE location != 'Москва' AND payment for services > 3

	name text
1	Кинап
2	ГАЗ
3	Рога и копыта
4	Станкостроительный зав

4) SELECT Professions.name, no, date, vacancies_number, payment, cost_of_hiring FROM Employment_contracts JOIN Professions ON Employment_contracts.profession_id = Professions.id JOIN Employers ON employment_contracts.employer_id = Employers.id WHERE last_job_place = location ORDER BY cost_of_hiring

Пояснение:

Не совсем понял, что нужно вывести в этом задании. Решил вывести все данные из таблицы Employment_contracts, кроме id. Также добавил название профессии и стоимость найма и отсортировал таблицу по ней.

	name text	no integer	date month	vacancies_number integer	payment integer	cost_of_hiring integer
1	Кровельщик	128	Февраль	2	20000	10000
2	Кровельщик	132	Апрель	1	10000	10000

8. UPDATE

UPDATE Employment_contracts
SET payment = payment - payment/100 * employers.benefit
FROM Employers WHERE employment_contracts.employer_id = Employers.id
SELECT payment FROM Employment_contracts

	payment integer
1	24500
2	18000
3	20000
4	27200
5	24500
6	9000
7	117600
8	44100
9	9800
10	98000
11	45000
12	17000
13	19600
14	14700
15	14700
16	24500
17	27200

9. Adding a column

- ALTER TABLE Employment_contracts ADD COLUMN bureau_earnig SERIAL
- 2) UPDATE Employment_contracts SET bureau_earning = payment * Recruitment_bureaus.payment_for_services/100 FROM Recruitment_bureaus WHERE employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id

	bureau_earning integer
1	735
2	540
3	2200
4	1088
5	735
6	270
7	3528
8	4851
9	294
10	3920
11	1350
12	1870
13	588
14	1323
15	1323
16	735
17	816

Level 2

10. IN, NOT IN

SELECT Recruitment_bureaus.no from Employment_contracts
 JOIN Employers ON employment_contracts.employer_id = Employers.id
 JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id
 WHERE location IN ('H. Новгород')

	no text	â
1	N5	
2	N4	

2) SELECT Professions.name FROM Professions WHERE name NOT IN(SELECT Professions.name FROM Employers JOIN Employment_contracts ON Employers.id =
Employment_contracts.employer_id
WHERE benefit < 10 AND professions.id = profession_id
AND Employers.id = employer_id)



3) SELECT date, Employment_contracts.no, Employers.name FROM Employment_contracts
JOIN Professions ON Employment_contracts.profession_id = Professions.id
JOIN Employers ON employment_contracts.employer_id = Employers.id
WHERE Professions.name IN ('Автоводитель') OR payment >= 14000;

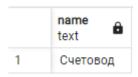
	date month	no integer	name text
1	Июнь	139	KPA3
2	Февраль	128	п/я 12687-у
3	Июнь	141	Станкостроительный зав
4	Июнь	140	Станкостроительный зав
5	Июнь	137	Рога и копыта
6	Май	134	Станкостроительный зав
7	Июнь	136	Кинап
8	Март	129	Рога и копыта
9	Май	133	KPA3
10	Июнь	142	Кинап
11	Апрель	131	Кинап
12	Январь	127	Станкостроительный зав
13	Сентябрь	143	ГАЗ
14	Июнь	138	Рога и копыта
15	Апрель	130	ГАЗ

4) SELECT Recruitment_bureaus.no, address FROM Employment_contracts JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id JOIN Employers ON Employment_contracts.employer_id = Employers.id WHERE benefit < 7 AND date NOT IN('Январь', 'Февраль')</p>

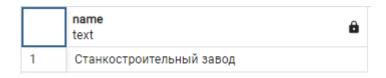
	no text	address text
1	N12	Киев
2	N6	Н. Новго
3	N4	Н. Москва
4	N12	Киев
5	N6	Н. Новго
6	N5	Н. Новго
7	N4	Н. Москва
8	N12	Киев
9	N6	Н. Новго
10	N8	Одесса
11	N8	Одесса
12	N4	Н. Москва

11. ALL-ANY

SELECT Professions.name FROM Employment_contracts
 JOIN Professions ON Employment_contracts.profession_id = Professions.id
 WHERE Employment_contracts.vacancies_number = ANY (
 SELECT MAX(vacancies_number) FROM Employment_contracts)

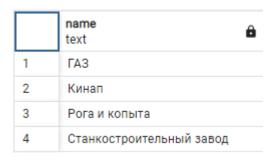


2) SELECT Employers.name FROM Employment_contracts JOIN Employers ON employment_contracts.employer_id = Employers.id JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id WHERE bureau_earning = ALL (SELECT MAX(bureau_earning) FROM Employment_contracts WHERE address != location)

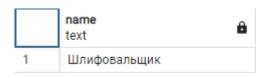


3) SELECT DISTINCT Employers.name FROM Employment_contracts

JOIN Recruitment_bureaus ON Employment_contracts.recruitment_bureau_id = Recruitment_bureaus.id



4) SELECT Professions.name FROM Employment_contracts
JOIN Professions ON Employment_contracts.profession_id = Professions.id
JOIN Employers ON Employment_contracts.employer_id = Employers.id
WHERE location = ALL(SELECT location FROM Employers
WHERE location = 'H. Новгород')



12. UNION

SELECT location from Employers
UNION
SELECT address from Recruitment_bureaus

	location text	
1	Киев	
2	Н. Новгород	
3	Кременчуг	
4	Н. Москва	
5	Саранск	
6	Приморск	
7	Одесса	

13. EXISTS

1) SELECT DISTINCT name FROM Professions

JOIN employment_contracts

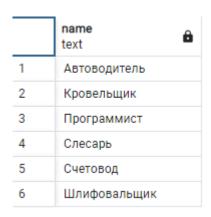
ON Professions.id = Employment_contracts.profession_id

WHERE EXISTS (

SELECT location FROM Employers

WHERE location != 'Приморск'

AND Employers.id = employer id)



2) SELECT Recruitment bureaus.no FROM Recruitment bureaus

JOIN Employment_contracts

ON Recruitment_bureaus.id = Employment_contracts.recruitment_bureau_id WHERE EXISTS (

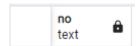
SELECT cost_of_hiring FROM Professions

WHERE Professions.id = profession id

AND cost of hiring = ALL(

SELECT cost of hiring FROM Professions

WHERE cost_of_hiring > 15000))



Необходимо добавить такое бюро найма, которое бы работала с профессиями, стоимость найма которых > 15000

3) SELECT no FROM Recruitment bureaus

WHERE NOT EXISTS(

SELECT * FROM Employment_contracts, Professions, Employers

WHERE Professions.id = Profession id

AND Employers.id = employer_id

AND Recruitment_bureaus.id = recruitment_bureau_id

AND location = last_job_place)

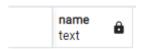
	no text	â
1	N5	
2	N12	
3	N8	

4) SELECT name FROM Employers

WHERE EXISTS (

SELECT payment FROM employment_contracts
WHERE Employers.id = employer_id AND date = 'Апрель'

AND payment > 100000)



Необходимо добавить договор, который был оформлен в апреле и его стоимость >= 100000.

14. Aggregate functions

1) SELECT Recruitment_bureaus.no, AVG(bureau_earning) FROM

Recruitment_bureaus

JOIN Employment_contracts

ON recruitment_bureau_id = Recruitment_bureaus.id

WHERE recruitment_bureau_id IN(

SELECT Recruitment bureaus.id

FROM Recruitment bureaus

WHERE NOT EXISTS(

SELECT 1 FROM Employers

WHERE location = 'Одесса'

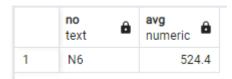
AND (Employers.id, Recruitment_bureaus.id)

NOT IN(

SELECT emloyer_id, recruitment_bureau_id

FROM Employment_contracts)))

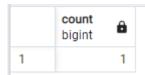
GROUP BY Recruitment bureaus.no



2) SELECT SUM(payment) FROM Employment_contracts



3) SELECT COUNT(DISTINCT Professions.name) FROM Professions JOIN Employment_contracts ON Professions.id = Employment_contracts.profession_id JOIN Employers ON employer_id = Employers.id WHERE date NOT IN ('Январь') AND location = 'H. Новгород'



4) SELECT Professions.name, AVG(vacancies_number) FROM Professions JOIN Employment_contracts ON Professions.id = profession_id JOIN employers ON employer_id = employers.id WHERE cost_of_hiring > 20000 GROUP BY Professions.name

	name text	â	avg numeric
1	Автоводитель		1.000000000000
2	Программист		3.000000000000
3	Счетовод		4.000000000000

15. GROUP BY

SELECT Employers.name, Recruitment_bureaus.no, SUM(payment)
 FROM Employment_contracts
 JOIN Employers ON employer_id = Employers.id
 JOIN Recruitment_bureaus
 ON Recruitment_bureaus.id = Recruitment_bureau_id
 GROUP BY Employers.name, Recruitment_bureaus.no

	name text	no text	sum bigint
1	ГАЗ	N4	27200
2	ГАЗ	N5	27200
3	Кинап	N4	24500
4	Кинап	N5	98000
5	Кинап	N6	24500
6	KPA3	N4	117600
7	KPA3	N6	19600
8	п/я 12687-у	N4	18000
9	п/я 12687-у	N6	9000
10	Рога и копы	N12	37000
11	Рога и копы	N4	45000
12	Станкостро	N12	44100
13	Станкостро	N6	34300
14	Станкостро	N8	29400

2) SELECT Recruitment_bureaus.no, COUNT(Employment_contracts.no) FROM Employment_contracts JOIN Recruitment_bureaus ON Recruitment_bureau_id = Recruitment_bureaus.id GROUP BY Recruitment_bureaus.no HAVING COUNT(Employment_contracts.no) > 2

	no text	â	count bigint	â
1	N12			3
2	N6			5
3	N4			5

3) SELECT location, COUNT(name) FROM Employers GROUP BY location

	location text	count bigint	â
1	Приморск		1
2	Н. Новго		1
3	Саранск		1
4	Одесса		2
5	Кременчуг		1

4) SELECT CAST(date AS TEXT) AS month_or_bureau, SUM (payment) FROM employment_contracts

GROUP BY date

HAVING SUM(payment) > 200000

UNION

SELECT Recruitment_bureaus.no AS month_or_bureau,

SUM(payment) FROM employment_contracts

JOIN Recruitment_bureaus

ON Recruitment_bureaus.id = recruitment_bureau_id

GROUP BY month_OR_bureau

HAVING SUM(payment) > 200000

	month_or_bureau text	sum bigint
1	N4	232300
2	Июнь	233500