Министерство образования Республики Беларусь

Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей

Кафедра электронных вычислительных машин

ОТЧЕТ
по лабораторной работе № 5
на тему
РЕАЛИЗАЦИЯ SQL-ЗАПРОСОВ НА ВЫБОРКУ С ГРУППИРОВАНИЕМ РЕЗУЛЬТАТОВ. «ШКОЛА»

Студент: А.Н. Климович

Преподаватель: Д.В. Куприянова

МИНСК 2024

1 ЦЕЛЬ РАБОТЫ

Формирование SQL-операторов для простой выборки данных из таблиц с использованием обобщающих функций и секций `GROUP BY`, `HAVING` и других.

2 ВЫПОЛНЕНИЕ РАБОТЫ

2.1 Порядок выполнения работы

1. Получить у преподавателя задания по собственной схеме данных, созданной в лабораторной работе №2 и реализованной в виде таблиц в СУБД в лабораторной работе №3. Создать запросы по заданиям (по одному запросу на каждое задание).

2. Правила выполнения заданий:

- для каждого задания создать реализацию в виде одного оператора SQL `SELECT`, в котором можно использовать подзапросы и группировку данных;
- обратить внимание, что использование скалярных (особенно соотнесенных!) подзапросов в предложении `SELECT` следует ограничить, т.к. они ухудшают производительность и анализ запроса, поэтому, если запрос затрагивает несколько таблиц, то сначала надо собрать данные с помощью соединения данных таблиц, и только потом выполнять их обработку (например, группировать);
- перед запуском запроса на выполнение, изучить данные в используемых запросом таблицах, и если требуется добавить новые данные, чтобы результат выборки не был пустым;
- выполнить запрос и проанализировать его результат – если есть расхождения между изученными данными и результатом запроса, то есть повод задуматься о проверке правильности выполнения этого задания.

2.2 Выполнение запросов

2.2.1 Запрос 1

Объединить и вывести имена и фамилии учеников и сотрудников в алфавитном порядке, у которых имя начинается на букву “И”. Для учеников также вывести номер и букву класса, к которому они принадлежат.

```
SELECT
    first_name,
    last_name,
    class_number,
    class_letter
FROM main_scheme.student
INNER JOIN main_scheme.class
ON student.class_id = main_scheme.class.id
WHERE first_name LIKE 'И%'
UNION
SELECT
    first_name,
    last_name,
    NULL,
    NULL
FROM main_scheme.employee
WHERE first_name LIKE 'И%'
ORDER BY last_name;
```

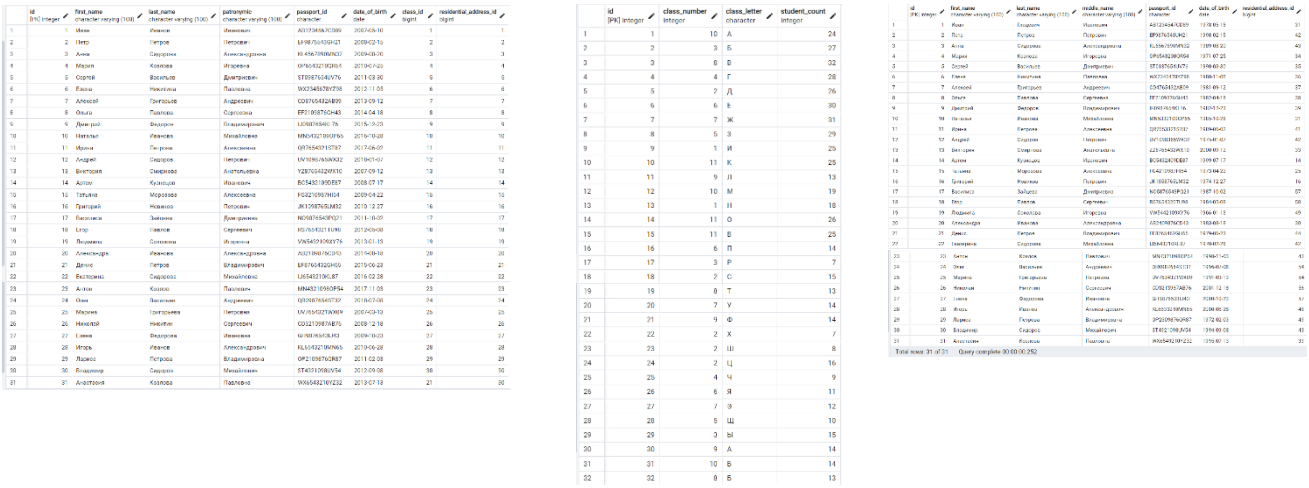


Рисунок 2.1 – Таблицы student, class и employee, участвующие в запросе 1

Query Query History

```
1  SELECT
2      first_name,
3      last_name,
4      class_number,
5      class_letter
6  FROM main_scheme.student
7  INNER JOIN main_scheme.class
8  ON student.class_id = main_scheme.class.id
9  WHERE first_name LIKE 'И%'
10 UNION
11 SELECT
12     first_name,
13     last_name,
14     NULL,
15     NULL
16 FROM main_scheme.employee
17 WHERE first_name LIKE 'И%'
18 ORDER BY last_name;
```

Data Output Messages Notifications

	first_name character varying (100)	last_name character varying (100)	class_number integer	class_letter character
1	Иван	Глецевич	[null]	[null]
2	Иван	Иванов	10	А
3	Игорь	Иванов	5	Щ
4	Игорь	Иванов	[null]	[null]
5	Ирина	Петрова	9	Л
6	Ирина	Петрова	[null]	[null]

Total rows: 6 of 6 Query complete 00:00:00.046

Рисунок 2.2 – Результат выполнения запроса 1

2.2.2 Запрос 2

Найти фамилии учеников и сотрудников школы, которые живут по одному адресу и имеют одну и ту же фамилию.

```
SELECT
    residential_address_id,
    last_name,
    residential_address.city,
    residential_address.street,
    residential_address.house_number
FROM main_scheme.employee
INNER JOIN main_scheme.residential_address
ON residential_address.id =
    employee.residential_address_id
INTERSECT
SELECT
    residential_address_id,
    last_name,
    residential_address.city,
    residential_address.street,
    residential_address.house_number
```

```
FROM main_scheme.student
INNER JOIN main_scheme.residential_address
ON residential_address_id =
    student.residential_address_id;
```

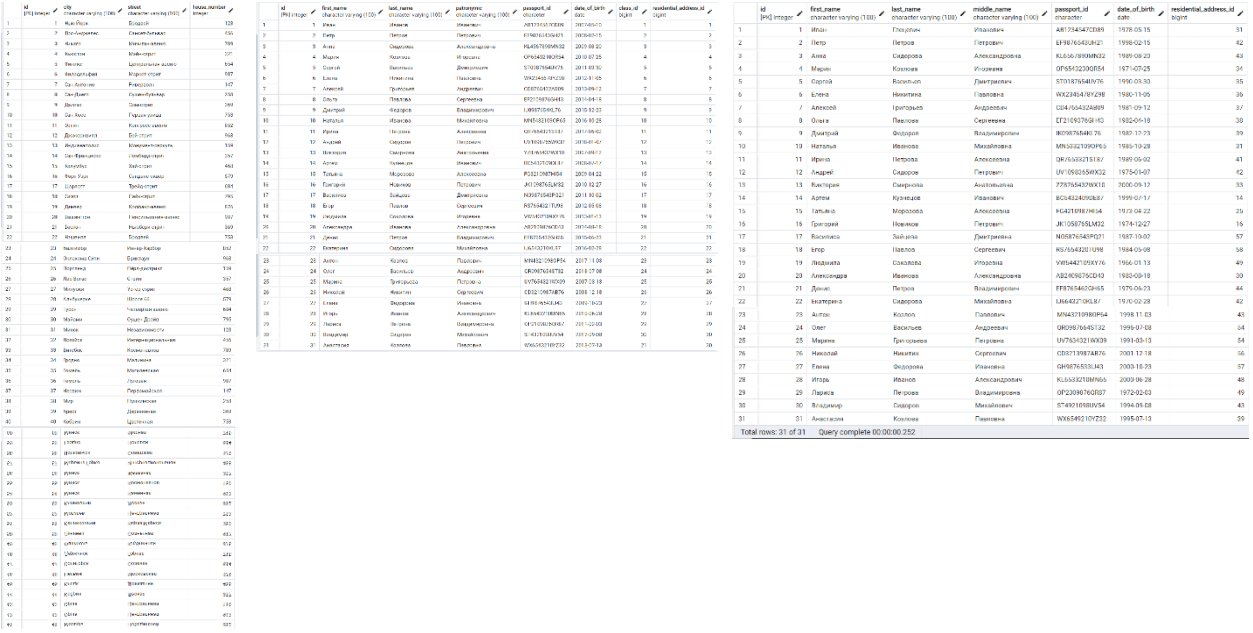


Рисунок 2.3 – Таблицы residential_address, student и employee, участвующие в запросе 2

Query Query History

```
SELECT
    residential_address_id,
    last_name,
    residential_address.city,
    residential_address.street,
    residential_address.house_number
FROM main_scheme.employee
    INNER JOIN main_scheme.residential_address
    ON residential_address.id = employee.residential_address_id
INTERSECT
SELECT
    residential_address_id,
    last_name,
    residential_address.city,
    residential_address.street,
    residential_address.house_number
FROM main_scheme.student
    INNER JOIN main_scheme.residential_address
    ON residential_address_id = student.residential_address_id;
```

Data Output Messages Notifications

	residential_address_id bigint	last_name character varying (100)	city character varying (100)	street character varying (100)	house_number integer
1	16	Новиков	Форт-Уэрт	Санданс-сквер	579
2	14	Кузнецов	Сан-Франциско	Ломбард-стрит	357

Рисунок 2.4 – Результат выполнения запроса 2

2.2.3 Запрос 3

Найти ФИО людей, которые есть в таблице “employee”, но нет в таблице “student”. При этом необходимо также вывести их город проживания.

```
WITH uniq_people AS
     (SELECT first_name, last_name, middle_name
      FROM main_scheme.student
      EXCEPT
      SELECT first_name, last_name, middle_name
      FROM main_scheme.employee)
SELECT first_name, last_name, middle_name, city
FROM main_scheme.student
INNER JOIN main_scheme.residential_address
ON student.residential_address_id =
    residential_address.id
WHERE first_name
IN(SELECT first_name FROM uniq_people);
```

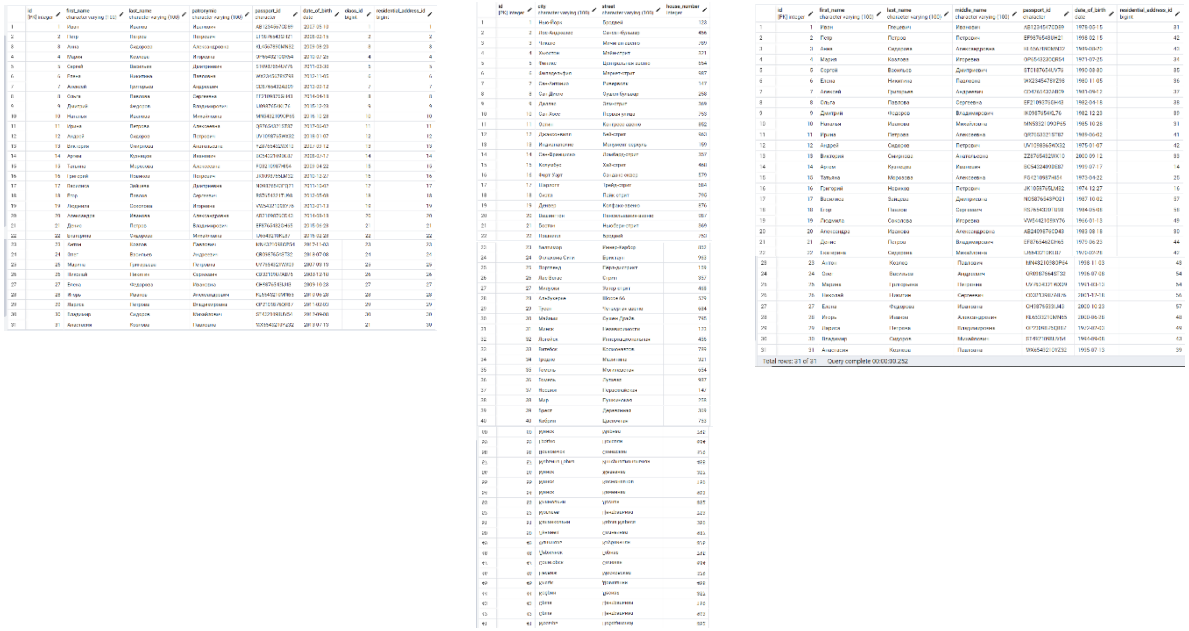


Рисунок 2.5 – Таблицы student, employee и residential_address, участвующие в запросе 3

```
1  WITH uniq_people AS
2      (SELECT first_name, last_name, middle_name
3       FROM main_scheme.student
4       EXCEPT
5       SELECT first_name, last_name, middle_name
6       FROM main_scheme.employee)
7  SELECT first_name, last_name, middle_name, city
8  FROM main_scheme.student
9  INNER JOIN main_scheme.residential_address
10 ON student.residential_address_id = residential_address.id
11 WHERE first_name IN(SELECT first_name FROM uniq_people);
```

Data Output Messages Notifications

	first_name character varying (100)	last_name character varying (100)	middle_name character varying (100)	city character varying (100)
1	Иван	Иванов	Иванович	Нью-Йорк
2	Артем	Кузнецов	Иванович	Сан-Франциско
3	Григорий	Новиков	Петрович	Форт-Уэрт

Рисунок 2.6 – Результат выполнения запроса 3

2.2.4 Запрос 4

Вывести название предмета, день недели, номер класса, общее количество учеников, для которых номер класса является общим, а также количество таких классов. При этом день недели должен быть понедельником, а номера классов должны иметь номер от 9 до 11.

```
WITH common_class_info AS
    (SELECT class_number,
        SUM(student_count) AS all_student_count,
        COUNT(class_number) AS all_classes
        FROM main_scheme.class
        GROUP BY class_number)
SELECT
    subject.name,
    schedule.week_day,
    common_class_info.class_number,
    common_class_info.all_student_count,
    common_class_info.all_classes
FROM main_scheme.subject
INNER JOIN main_scheme.schedule
ON main_scheme.schedule.subject_id =
    main_scheme.subject.id
INNER JOIN main_scheme.class
ON main_scheme.class.id =
    main_scheme.schedule.class_id
INNER JOIN common_class_info
ON common_class_info.class_number =
```

```
        main_scheme.class.class_number
    WHERE common_class_info.class_number
        BETWEEN 9 AND 11
        AND schedule.week_day = 'Понедельник';
```

	id [PK] integer	class_number integer	class_letter character	student_count integer
1	1	10	А	24
2	2	3	Б	27
3	3	8	В	32
4	4	4	Г	28
5	5	2	Д	26
6	6	6	Е	30
7	7	7	Ж	31
8	8	5	З	29
9	9	1	И	25
10	10	11	К	25
11	11	9	Л	13
12	12	10	М	19
13	13	1	Н	18
14	14	11	О	26
15	15	11	В	25
16	16	6	П	14
17	17	3	Р	7
18	18	2	С	15
19	19	8	Т	13
20	20	7	У	14
21	21	9	Ф	14
22	22	2	Х	7
23	23	2	Ш	8
24	24	2	Ц	16
25	25	4	Ч	9
26	26	6	Я	11
27	27	7	Э	12
28	28	5	Щ	10
29	29	3	Ы	15
30	30	9	А	14
31	31	10	Б	14
32	32	8	Б	13

	id [PK] integer	week_day character varying (20)	class_id bigint	subject_id bigint
1	1	Понедельник	1	1
2	2	Понедельник	2	2
3	3	Вторник	3	3
4	4	Среда	4	4
5	5	Понедельник	5	5
6	6	Пятница	6	6
7	7	Пятница	7	7
8	8	Четверг	8	8
9	9	Среда	9	9
10	10	Вторник	10	10
11	11	Вторник	11	11
12	12	Четверг	12	12
13	13	Понедельник	13	13
14	14	Пятница	14	14
15	15	Четверг	15	15
16	16	Среда	16	16
17	17	Понедельник	17	17
18	18	Понедельник	18	18
19	19	Понедельник	19	19
20	20	Вторник	20	20
21	21	Понедельник	21	21
22	22	Среда	22	22
23	23	Понедельник	23	23
24	24	Вторник	24	24
25	25	Понедельник	25	25
26	26	Четверг	26	26
27	27	Понедельник	27	27
28	28	Понедельник	28	28
29	29	Среда	29	29
30	30	Понедельник	30	30

	id [PK] integer	name character varying (100)	hours integer	classroom integer
1	1	Математика	60	101
2	2	Физика	45	102
3	3	История	30	103
4	4	Литература	45	104
5	5	Биология	60	105
6	6	Химия	45	106
7	7	География	30	107
8	8	Английский язык	60	108
9	9	Французский язык	45	109
10	10	Немецкий язык	30	110
11	11	Испанский язык	45	111
12	12	Информатика	60	112
13	13	Физкультура	30	113
14	14	Музыка	45	114
15	15	ИЗО	60	115
16	16	Трудовое обучение	30	116
17	17	Обществознание	45	117
18	18	Право	60	118
19	19	Экономика	30	119
20	20	Психология	45	120
21	21	Философия	60	121
22	22	Религиоведение	30	122
23	23	Хореография	45	123
24	24	Театральное искусство	60	124
25	25	Основы медицины	30	125
26	26	Дизайн	45	126
27	27	Астрономия	60	127
28	28	Геология	30	128
29	29	Архитектура	45	129
30	30	Лингвистика	60	130

Рисунок 2.7 – Таблицы class, schedule и subject, участвующие в запросе 4

Query Query History

```
1  WITH common_class_info AS
2      (SELECTclass_number,
3          SUM(student_count) AS all_student_count,
4          COUNT(class_number) AS all_classes
5      FROM main_scheme.class
6      GROUP BY class_number)
7  SELECT
8      subject.name,
9      schedule.week_day,
10     common_class_info.class_number,
11     common_class_info.all_student_count,
12     common_class_info.all_classes
13 FROM main_scheme.subject
14 INNER JOIN main_scheme.schedule
15 ON main_scheme.schedule.subject_id = main_scheme.subject.id
16 INNER JOIN main_scheme.class
17 ON main_scheme.class.id = main_scheme.schedule.class_id
18 INNER JOIN common_class_info
19 ON common_class_info.class_number = main_scheme.class.class_number
20 WHERE common_class_info.class_number BETWEEN 9 AND 11
21     AND schedule.week_day = 'Понедельник';
```

Data Output Messages Notifications

	name character varying (100)	week_day character varying (20)	class_number integer	all_student_count bigint	all_classes bigint
1	Математика	Понедельник	10	57	3
2	Философия	Понедельник	9	41	3
3	Лингвистика	Понедельник	9	41	3

Рисунок 2.8 – Результат выполнения запроса 4

2.2.5 Запрос 5

Вывести имя и фамилию ученика, номер класса и оценку, полученную в промежуток времени от 01.01.2023 до 01.05.2023, которая больше или равна средней оценки среди всех оценок.

```
SELECT
    student.last_name,
    student.first_name,
    class.class_number,
    gradebook.grade
FROM main_scheme.student
LEFT OUTER JOIN main_scheme.class
    ON class.id = student.class_id
LEFT OUTER JOIN main_scheme.gradebook
    ON gradebook.student_id = student.id
WHERE gradebook.grade_date
BETWEEN '2023-01-01' AND '2023-05-01'
AND grade >=
    (SELECT ROUND(AVG(grade), 1) AS avg_grade
     FROM main_scheme.gradebook);
```

	id [PK] integer	employee_id bigint	student_id bigint	subject_id bigint	grade_date date	grade smallint
1	1	1	1	1	2023-01-15	8
2	2	2	2	2	2023-02-20	7
3	3	3	3	3	2023-03-25	9
4	4	4	4	1	2023-04-30	6
5	5	5	5	2	2023-05-05	8
6	6	6	6	3	2023-06-10	7
7	7	7	7	1	2023-07-15	9
8	8	8	8	2	2023-08-20	5
9	9	9	9	3	2023-09-25	8
10	10	1	1	1	2023-10-30	7
11	11	1	1	2	2023-11-05	6
12	12	1	2	3	2023-12-10	9
13	13	3	3	1	2024-01-15	8
14	14	4	4	2	2024-02-20	7
15	15	5	5	3	2024-03-25	9
16	16	6	6	1	2024-04-30	6
17	17	7	7	2	2024-05-05	8
18	18	8	8	3	2024-06-10	7
19	19	1	9	1	2024-07-15	9
20	20	2	2	2	2024-08-20	5
21	21	2	2	3	2024-09-25	8
22	22	2	2	1	2024-10-30	7
23	23	2	3	2	2024-11-05	6
24	24	4	4	3	2024-12-10	9
25	25	5	12	1	2023-01-15	8
26	26	6	12	2	2023-02-20	7
27	27	7	17	3	2023-03-25	9
28	28	8	18	1	2023-04-30	6
29	29	9	19	2	2023-05-05	8
30	30	3	13	3	2023-06-10	7

	id [PK] integer	first_name character varying (100)	last_name character varying (100)	patronymic character varying (100)	passport_id character	date_of_birth date	class_id bigint	residential_address_id bigint
1	1	Иван	Иванов	Иванович	[illegible]	2007-05-15	1	1
2	2	Петр	Петров	Петрович	[illegible]	2008-07-15	2	2
3	3	Алекс	Сидоров	Александрович	[illegible]	2009-08-20	3	3
4	4	Мария	Иванова	Игоревна	[illegible]	2010-07-05	4	4
5	5	Сергей	Васильев	Дмитриевич	[illegible]	2010-03-20	5	5
6	6	Елена	Иванова	Павловна	[illegible]	2010-11-05	6	6
7	7	Алексей	Григорьев	Андреевич	[illegible]	2010-06-12	7	7
8	8	Ольга	Павлова	Сергеевна	[illegible]	2010-04-18	8	8
9	9	Дмитрий	Смирнов	Владимирович	[illegible]	2010-03-23	9	9
10	10	Наталья	Иванова	Михайловна	[illegible]	2010-10-20	10	10
11	11	Ирина	Петрова	Алексеевна	[illegible]	2010-06-02	11	11
12	12	Андрей	Сидоров	Петрович	[illegible]	2010-07-07	12	12
13	13	Виктория	Смирнова	Анатольевна	[illegible]	2010-04-13	13	13
14	14	Артем	Иванов	Иванович	[illegible]	2010-07-17	14	14
15	15	Татьяна	Морозова	Алексеевна	[illegible]	2010-04-27	15	15
16	16	Григорий	Иванов	Петрович	[illegible]	2010-12-27	16	16
17	17	Наталья	Зайцева	Дмитриевна	[illegible]	2010-10-07	17	17
18	18	Петр	Павлов	Сергеевич	[illegible]	2010-09-08	18	18
19	19	Екатерина	Соколова	Игоревна	[illegible]	2010-01-13	19	19
20	20	Александра	Иванова	Александровна	[illegible]	2010-08-13	20	20
21	21	Денис	Петров	Владимирович	[illegible]	2010-06-23	21	21
22	22	Екатерина	Сидорова	Михайловна	[illegible]	2010-02-28	22	22
23	23	Антон	Новиков	Павлович	[illegible]	2010-11-03	23	23
24	24	Олег	Васильев	Андреевич	[illegible]	2010-07-03	24	24
25	25	Марина	Григорьева	Петровна	[illegible]	2012-02-13	25	25
26	26	Евгений	Иванов	Сергеевич	[illegible]	2008-12-18	26	26
27	27	Елена	Федорова	Ивановна	[illegible]	2009-10-23	27	27
28	28	Игорь	Иванов	Александрович	[illegible]	2010-06-28	28	28
29	29	Лариса	Петрова	Владимировна	[illegible]	2011-02-03	29	29
30	30	Владимир	Сидоров	Михайлович	[illegible]	2012-09-08	30	30
31	31	Анастасия	Иванова	Павловна	[illegible]	2013-07-13	31	30

	id [PK] integer	class_number integer	class_letter character	student_count integer
1	1	10	А	24
2	2	3	Б	27
3	3	8	В	32
4	4	4	Г	28
5	5	2	Д	26
6	6	6	Е	30
7	7	7	Ж	31
8	8	5	З	29
9	9	1	И	25
10	10	11	К	25
11	11	9	Л	13
12	12	10	М	19
13	13	1	Н	18
14	14	11	О	26
15	15	11	В	25
16	16	6	П	14
17	17	3	Р	7
18	18	2	С	15
19	19	8	Т	13
20	20	7	У	14
21	21	9	Ф	14
22	22	2	Х	7
23	23	2	Ш	8
24	24	2	Ц	16
25	25	4	Ч	9
26	26	6	Я	11
27	27	7	Э	12
28	28	5	Щ	10
29	29	3	Ы	15
30	30	9	А	14
31	31	10	Б	14
32	32	8	Б	13

Рисунок 2.9 – Таблицы gradebook, student и class, участвующие в запросе 5

Query Query History

```
1  SELECT
2      student.last_name,
3      student.first_name,
4      class.class_number,
5      gradebook.grade
6  FROM main_scheme.student
7  LEFT OUTER JOIN main_scheme.class
8  ON class.id = student.class_id
9  LEFT OUTER JOIN main_scheme.gradebook
10 ON gradebook.student_id = student.id
11 WHERE gradebook.grade_date BETWEEN '2023-01-01' AND '2023-05-01'
12     AND grade >= (SELECT ROUND(AVG(grade), 1) AS avg_grade
13                   FROM main_scheme.gradebook);
```

Data Output Messages Notifications

	last_name character varying (100)	first_name character varying (100)	class_number integer	grade smallint
1	Иванов	Иван	10	8
2	Сидорова	Анна	8	9
3	Сидоров	Андрей	10	8
4	Зайцева	Василиса	3	9

Рисунок 2.10 – Результат выполнения запроса 5

2.2.6 Запрос 6

Найти ФИО сотрудников, название предмета, который они знают, и соответствующее количество часов предмета, которое должно равняться максимальному количеству часов среди всех предметов.

```
SELECT
    subject.name,
    subject.hours,
    employee.last_name,
    employee.first_name,
    employee.middle_name
FROM main_scheme.subject
FULL OUTER JOIN main_scheme.knowledge_of_subject
ON knowledge_of_subject.subject_id = subject.id
FULL OUTER JOIN main_scheme.employee
ON employee.id = knowledge_of_subject.employee_id
WHERE subject.hours =
    (SELECT MAX(hours) AS max_hours
     FROM main_scheme.subject);
```

	id [PK] integer	name character varying (100)	hours integer	classroom integer
1	1	Математика	60	101
2	2	Физика	45	102
3	3	История	30	103
4	4	Литература	45	104
5	5	Биология	60	105
6	6	Химия	45	106
7	7	География	30	107
8	8	Английский язык	60	108
9	9	Французский язык	45	109
10	10	Немецкий язык	30	110
11	11	Испанский язык	45	111
12	12	Информатика	60	112
13	13	Физкультура	30	113
14	14	Музыка	45	114
15	15	ИЗО	60	115
16	16	Трудовое обучение	30	116
17	17	Обществознание	45	117
18	18	Право	60	118
19	19	Экономика	30	119
20	20	Психология	45	120
21	21	Философия	60	121
22	22	Религиоведение	30	122
23	23	Хореография	45	123
24	24	Театральное искусство	60	124
25	25	Основы медицины	30	125
26	26	Дизайн	45	126
27	27	Астрономия	60	127
28	28	Геология	30	128
29	29	Архитектура	45	129
30	30	Лингвистика	60	130

	employee_id [PK] bigint	subject_id bigint
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30

	id [PK] integer	first_name character varying (100)	last_name character varying (100)	middle_name character varying (100)	passport_id character	date_of_birth date	residential_address_id bigint
1	1	Иван	Глецевич	Иванович	AB1234547CD89	1978-05-15	31
2	2	Петр	Петров	Петрович	EF9876543UH21	1998-02-15	42
3	3	Анна	Сидорова	Александровна	KL6567890MN32	1989-08-20	43
4	4	Мария	Козлова	Игоревна	OP6543230QR54	1971-07-25	34
5	5	Сергей	Васильев	Дмитриевич	ST0187654UV76	1990-03-30	35
6	6	Елена	Никитина	Павловна	WX2345478YZ98	1980-11-05	36
7	7	Алексей	Григорьев	Андреевич	CD4765432AB09	1981-09-12	37
8	8	Ольга	Павлова	Сергеевна	EF2109376GH43	1982-04-18	38
9	9	Дмитрий	Федоров	Владимирович	IK0987654KL76	1982-12-23	39
10	10	Наталья	Иванова	Михайловна	MN53321090P65	1985-10-28	31
11	11	Ирина	Петрова	Алексеевна	QR7653321ST87	1989-06-02	41
12	12	Андрей	Сидоров	Петрович	UV1098365WX32	1975-01-07	42
13	13	Виктория	Смирнова	Анатольевна	ZZ8765432WX10	2000-09-12	33
14	15	Татьяна	Морозова	Алексеевна	FG4210987HI54	1973-04-22	25
15	17	Василиса	Зайцева	Дмитриевна	NO5876543PQ21	1987-10-02	57
16	18	Егор	Павлов	Сергеевич	RS7654320TU98	1984-05-08	58
17	19	Людмила	Соколова	Игоревна	VW5442109XY76	1965-01-13	49
18	20	Александра	Иванова	Александровна	AB2409876CD43	1983-08-18	30
19	21	Денис	Петров	Владимирович	EF8765462GH65	1979-06-23	44
20	22	Екатерина	Сидорова	Михайловна	IJ6643210KL87	1970-02-28	42
21	23	Антон	Козлов	Павлович	MN4321098OP64	1998-11-03	43
22	24	Олег	Васильев	Андреевич	QR0987664ST32	1996-07-08	54
23	25	Марина	Григорьева	Петровна	UV7634321WX09	1991-03-13	54
24	26	Николай	Никитин	Сергеевич	CD3213987AB76	2001-12-18	56
25	27	Елена	Федорова	Ивановна	GH9876533IJ43	2000-10-23	57
26	28	Игорь	Иванов	Александрович	KL6533210MN65	2000-06-28	48
27	29	Лариса	Петрова	Владимировна	OP23098765QR87	1972-02-03	49
28	30	Владимир	Сидоров	Михайлович	ST4921098UV54	1994-09-08	43
29	31	Анастасия	Козлова	Павловна	WX6549210YZ32	1995-07-13	39
30	14	Алексей	Кузнецов	Сергеевич	BC54324090E87	1999-07-17	14
31	16	Дмитрий	Новиков	Иванович	JK1058765LM32	1974-12-27	16

Рисунок 2.11 – Таблицы subject, knowledge_of_subject и employee, участвующие в запросе 6

Query Query History

```
SELECT
    subject.name,
    subject.hours,
    employee.last_name,
    employee.first_name,
    employee.middle_name
FROM main_scheme.subject
FULL OUTER JOIN main_scheme.knowledge_of_subject
ON knowledge_of_subject.subject_id = subject.id
FULL OUTER JOIN main_scheme.employee
ON employee.id = knowledge_of_subject.employee_id
WHERE subject.hours = (SELECT MAX(hours) AS max_hours
FROM main_scheme.subject);
```

Data Output Messages Notifications

	name character varying (100)	hours integer	last_name character varying (100)	first_name character varying (100)	middle_name character varying (100)
1	Математика	60	Глецевич	Иван	Иванович
2	Биология	60	Васильев	Сергей	Дмитриевич
3	Английский язык	60	Павлова	Ольга	Сергеевна
4	Информатика	60	Сидоров	Андрей	Петрович
5	ИЗО	60	Морозова	Татьяна	Алексеевна
6	Право	60	Павлов	Егор	Сергеевич
7	Философия	60	Петров	Денис	Владимирович
8	Театральное искусство	60	Васильев	Олег	Андреевич
9	Астрономия	60	Федорова	Елена	Ивановна
10	Лингвистика	60	Сидоров	Владимир	Михайлович

Рисунок 2.12 – Результат выполнения запроса 6

2.2.7 Запрос 7

Найти ФИО сотрудников, название должности и процентную ставку, которая больше медианной ставки среди ставок.

```
SELECT
    employee.last_name,
    employee.first_name,
    employee.middle_name,
    job_position.name,
    job_position.rate
FROM main_scheme.employee
INNER JOIN main_scheme.employee_position
ON employee_position.employee_id = employee.id
INNER JOIN main_scheme.job_position
ON job_position.id =
    employee_position.job_position_id
WHERE main_scheme.job_position.rate >
    (SELECT PERCENTILE_CONT(0.5)
        WITHIN GROUP(ORDER BY rate) AS median_rate
     FROM main_scheme.job_position);
```

	id [PK] integer	name character varying (100)	rate real	salary money
1	1	Директор	1	100 000,00 ?
2	2	Заместитель директора	0.9	90 000,00 ?
3	3	Учитель математики	0.8	80 000,00 ?
4	4	Учитель русского языка	0.8	80 000,00 ?
5	5	Учитель физики	0.8	80 000,00 ?
6	6	Учитель химии	0.8	80 000,00 ?
7	7	Учитель биологии	0.8	80 000,00 ?
8	8	Учитель истории	0.8	80 000,00 ?
9	9	Учитель географии	0.8	80 000,00 ?
10	10	Учитель английского языка	0.8	80 000,00 ?
11	11	Учитель физической культуры	0.8	80 000,00 ?
12	12	Учитель искусства	0.8	80 000,00 ?
13	13	Учитель музыки	0.8	80 000,00 ?
14	14	Учитель информатики	0.8	80 000,00 ?
15	15	Учитель технологии	0.8	80 000,00 ?
16	16	Учитель начальных классов	0.8	80 000,00 ?
17	17	Педагог-психолог	0.7	70 000,00 ?
18	18	Социальный педагог	0.7	70 000,00 ?
19	19	Логопед	0.7	70 000,00 ?
20	20	Библиотекарь	0.7	70 000,00 ?
21	21	Администратор	0.6	60 000,00 ?
22	22	Секретарь	0.6	60 000,00 ?
23	23	Бухгалтер	0.6	60 000,00 ?
24	24	Сторож	0.5	50 000,00 ?
25	25	Уборщик	0.5	50 000,00 ?
26	26	Воспитатель	0.7	70 000,00 ?
27	27	Медсестра	0.6	60 000,00 ?
28	28	Дежурный по физической культуре	0.7	70 000,00 ?
29	29	Дежурный по охране	0.6	60 000,00 ?
30	30	Дежурный по уборке	0.5	50 000,00 ?

	id [PK] integer	first_name character varying (100)	last_name character varying (100)	middle_name character varying (100)	passport_id character	date_of_birth date	residential_address_id bigint
1	1	Иван	Глецевич	Иванович	AB1234547CD89	1978-05-15	31
2	2	Петр	Петров	Петрович	EF9876543UH21	1998-02-15	42
3	3	Анна	Сидорова	Александровна	KL6567890MN32	1989-08-20	43
4	4	Мария	Козлова	Игоревна	OP6543230QR54	1971-07-25	34
5	5	Сергей	Васильев	Дмитриевич	ST0187654UV76	1990-03-30	35
6	6	Елена	Никитина	Павловна	WX2345478YZ98	1980-11-05	36
7	7	Алексей	Григорьев	Андреевич	CD4765432AB09	1981-09-12	37
8	8	Ольга	Павлова	Сергеевна	EF2109376GH43	1982-04-18	38
9	9	Дмитрий	Федоров	Владимирович	IK0987654KL76	1982-12-23	39
10	10	Наталья	Иванова	Михайловна	MN5332109OP65	1985-10-28	31
11	11	Ирина	Петрова	Алексеевна	QR7653321ST87	1989-06-02	41
12	12	Андрей	Сидоров	Петрович	UV1098365WX32	1975-01-07	42
13	13	Виктория	Смирнова	Анатольевна	ZZ8765432WX10	2000-09-12	33
14	15	Татьяна	Морозова	Алексеевна	FG4210987HI54	1973-04-22	25
15	17	Василиса	Зайцева	Дмитриевна	NO5876543PQ21	1987-10-02	57
16	18	Егор	Павлов	Сергеевич	RS7654320TU98	1984-05-08	58
17	19	Людмила	Соколова	Игоревна	VW5442109XY76	1966-01-13	49
18	20	Александра	Иванова	Александровна	AB2409876CD43	1983-08-18	30
19	21	Денис	Петров	Владимирович	EF8765462GH65	1979-06-23	44
20	22	Екатерина	Сидорова	Михайловна	IJ6643210KL87	1970-02-28	42
21	23	Антон	Козлов	Павлович	MN4321098OP64	1998-11-03	43
22	24	Олег	Васильев	Андреевич	QR0987664ST32	1996-07-08	54
23	25	Марина	Григорьева	Петровна	UV7634321WX09	1991-03-13	54
24	26	Николай	Никитин	Сергеевич	CD3213987AB76	2001-12-18	56
25	27	Елена	Федорова	Ивановна	GH9876533IJ43	2000-10-23	57
26	28	Игорь	Иванов	Александрович	KL6533210MN65	2000-06-28	48
27	29	Лариса	Петрова	Владимировна	OP2309876QR87	1972-02-03	49
28	30	Владимир	Сидоров	Михайлович	ST4921098UV54	1994-09-08	43
29	31	Анастасия	Козлова	Павловна	WX6549210YZ32	1995-07-13	39
30	14	Алексей	Кузнецов	Сергеевич	BC5432409DE87	1999-07-17	14
31	16	Дмитрий	Новиков	Иванович	JK1058765LM32	1974-12-27	16

	employee_id [FK] bigint	job_position_id bigint
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30

Рисунок 2.13 – Таблицы job_position, employee и employee_position, участвующие в запросе 7

Query Query History

```
SELECT
    employee.last_name,
    employee.first_name,
    employee.middle_name,
    job_position.name,
    job_position.rate
FROM main_scheme.employee
INNER JOIN main_scheme.employee_position
ON employee_position.employee_id = employee.id
INNER JOIN main_scheme.job_position
ON job_position.id = employee_position.job_position_id
WHERE main_scheme.job_position.rate >
    (SELECT PERCENTILE_CONT(0.5)
        WITHIN GROUP(ORDER BY rate) AS median_rate
     FROM main_scheme.job_position);
```

Data Output Messages Notifications

	last_name character varying (100)	first_name character varying (100)	middle_name character varying (100)	name character varying (100)	rate real
1	Глецевич	Иван	Иванович	Директор	1
2	Петров	Петр	Петрович	Заместитель директора	0.9

Рисунок 2.14 – Результат выполнения запроса 7

2.2.8 Запрос 8

Найти имя и фамилию учеников, их класс, а также предмет, по которому они получили минимальную оценку среди всех оценок. Кроме того, необходимо вывести дату этой оценки и фамилию учителя, который выставил оценку.

```
SELECT
    student.last_name AS student_last_name,
    student.first_name AS student_first_name,
    main_scheme.class.class_number,
    main_scheme.class.class_letter,
    subject.name,
    employee.last_name AS teacher_last_name,
    gradebook.grade,
    gradebook.grade_date
FROM main_scheme.student
INNER JOIN main_scheme.class
    ON class.id = student.class_id
INNER JOIN main_scheme.gradebook
    ON gradebook.student_id = student.id
INNER JOIN main_scheme.employee
    ON gradebook.employee_id = employee.id
INNER JOIN main_scheme.subject
    ON subject.id = gradebook.subject_id
WHERE main_scheme.gradebook.grade =
    (SELECT MIN(grade) AS min_grade
     FROM main_scheme.gradebook);
```

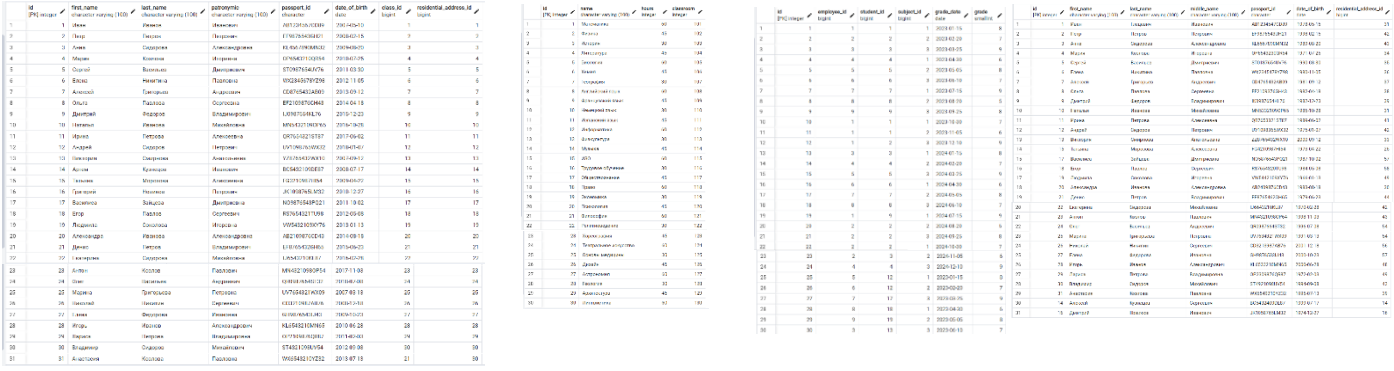


Рисунок 2.15 – Таблицы student, gradebook, subject и employee, участвующие в запросе 8

Query Query History

```
1  SELECT
2      student.last_name AS student_last_name,
3      student.first_name AS student_first_name,
4      main_scheme.class.class_number,
5      main_scheme.class.class_letter,
6      subject.name,
7      employee.last_name AS teacher_last_name,
8      gradebook.grade,
9      gradebook.grade_date
10 FROM main_scheme.student
11 INNER JOIN main_scheme.class
12 ON class.id = student.class_id
13 INNER JOIN main_scheme.gradebook
14 ON gradebook.student_id = student.id
15 INNER JOIN main_scheme.employee
16 ON gradebook.employee_id = employee.id
17 INNER JOIN main_scheme.subject
18 ON subject.id = gradebook.subject_id
19 WHERE main_scheme.gradebook.grade =
20     (SELECT MIN(grade) AS min_grade
21      FROM main_scheme.gradebook);
```

Data Output Messages Notifications

	student_last_name character varying (100)	student_first_name character varying (100)	class_number integer	class_letter character	name character varying (100)	teacher_last_name character varying (100)	grade smallint	grade_date date
1	Павлова	Ольга	5	З	Физика	Капуста	5	2023-08-20
2	Петров	Петр	3	Б	Физика	Дроздов	5	2024-08-20

Data saved successfully.

Рисунок 2.16 – Результат выполнения запроса 8

2.2.9 Запрос 9

Вывести имя и фамилию людей, город их проживания, а также id класса (если этот человек является учеников школы). Эти люди должны проживать в таких городах, где живут ещё хотя бы два человека, но не по его адресу.

```
SELECT
     student.last_name, student.first_name,
     residential_address.city, student.class_id
FROM main_scheme.student
LEFT OUTER JOIN main_scheme.residential_address
     ON residential_address.id =
     student.residential_address_id
WHERE residential_address.city IN
     (SELECT city
      FROM main_scheme.residential_address
      GROUP BY city HAVING COUNT(city) > 2)
UNION SELECT
     employee.first_name, employee.last_name,
     residential_address.city, NULL
FROM main_scheme.employee
LEFT OUTER JOIN main_scheme.residential_address
     ON residential_address.id =
     employee.residential_address_id
WHERE residential_address.city IN
     (SELECT city
      FROM main_scheme.residential_address
      GROUP BY city HAVING COUNT(city) > 2);
```

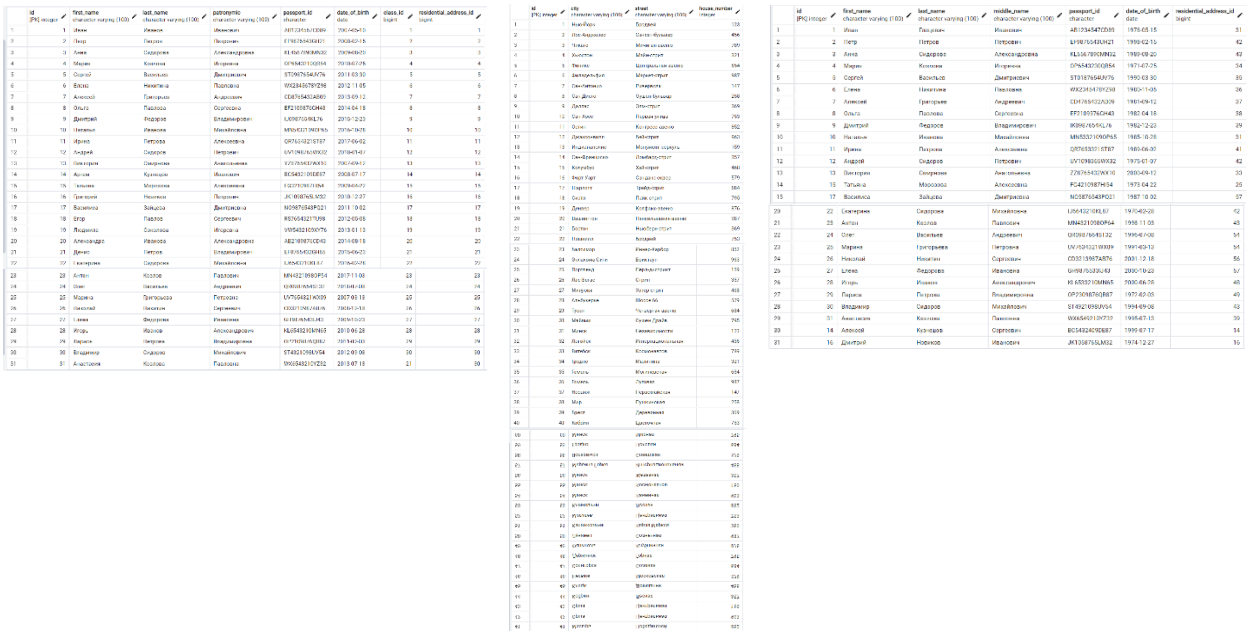


Рисунок 2.17 – Таблицы student, residential_address и employee, участвующие в запросе 9

Query Query History

```
1  SELECT
2      student.last_name, student.first_name,
3      residential_address.city, student.class_id
4  FROM main_scheme.student
5  LEFT OUTER JOIN main_scheme.residential_address
6  ON residential_address.id = student.residential_address_id
7  WHERE residential_address.city IN
8      (SELECT city FROM main_scheme.residential_address
9       GROUP BY city HAVING COUNT(city) > 2)
10 UNION
11 SELECT
12     employee.first_name, employee.last_name,
13     residential_address.city, NULL
14 FROM main_scheme.employee
15 LEFT OUTER JOIN main_scheme.residential_address
16 ON residential_address.id = employee.residential_address_id
17 WHERE residential_address.city IN
18     (SELECT city FROM main_scheme.residential_address
19      GROUP BY city HAVING COUNT(city) > 2);
```

Data Output Messages Notifications

	last_name character varying (100)	first_name character varying (100)	city character varying (100)	class_id bigint
1	Олег	Васильев	Минск	[null]
2	Иван	Глецевич	Минск	[null]
3	Марина	Григорьева	Минск	[null]
4	Наталья	Иванова	Минск	[null]
5	Николай	Никитин	Минск	[null]

Рисунок 2.18 – Результат выполнения запроса 9

2.2.10 Запрос 10

Вывести расписание занятия для 11-х классов (день недели, фамилия учителя, название предмета, номер кабинета, номер класса, общее количество учеников 11-х классов).

```
WITH common_class_info AS
     (SELECT SUM(student_count) AS all_students,
      class_number FROM main_scheme.class
      GROUP BY main_scheme.class.class_number)
SELECT
     schedule.week_day,
     employee.last_name,
     subject.name, subject.classroom,
     class.class_number,
     common_class_info.all_students
          AS count_of_students
FROM main_scheme.class
INNER JOIN main_scheme.schedule
     ON schedule.class_id = main_scheme.class.id
INNER JOIN main_scheme.subject
     ON schedule.subject_id = subject.id
INNER JOIN main_scheme.knowledge_of_subject
     ON knowledge_of_subject.subject_id = subject.id
INNER JOIN main_scheme.employee
     ON employee.id =
          knowledge_of_subject.employee_id
INNER JOIN common_class_info
     ON main_scheme.class.class_number =
          common_class_info.class_number
WHERE main_scheme.class.class_number = 11;
```

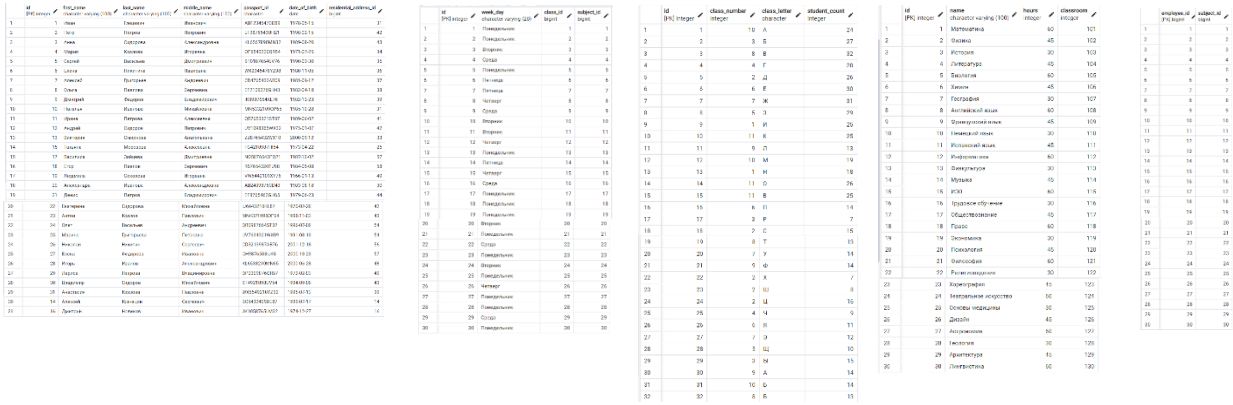


Рисунок 2.19 – Таблицы employee, schedule, class, subject и knowledge_of_subject, участвующие в запросе 10

Query Query History

```
1  WITH common_class_info AS
2      (SELECT SUM(student_count) AS all_students, class_number FROM main_scheme.class
3       GROUP BY main_scheme.class.class_number)
4  SELECT
5      schedule.week_day,
6      employee.last_name,
7      subject.name, subject.classroom,
8      class.class_number,
9      common_class_info.all_students AS count_of_students
10 FROM main_scheme.class
11 INNER JOIN main_scheme.schedule
12 ON schedule.class_id = main_scheme.class.id
13 INNER JOIN main_scheme.subject
14 ON schedule.subject_id = subject.id
15 INNER JOIN main_scheme.knowledge_of_subject
16 ON knowledge_of_subject.subject_id = subject.id
17 INNER JOIN main_scheme.employee
18 ON employee.id = knowledge_of_subject.employee_id
19 INNER JOIN common_class_info
20 ON main_scheme.class.class_number = common_class_info.class_number
21 WHERE main_scheme.class.class_number = 11;
```

Data Output Messages Notifications

	week_day character varying (20)	last_name character varying (100)	name character varying (100)	classroom integer	class_number integer	count_of_students bigint
1	Вторник	Иванова	Немецкий язык	110	11	76
2	Пятница	Кузнецов	Музыка	114	11	76
3	Четверг	Морозова	ИЗО	115	11	76

Рисунок 2.20 – Результат выполнения запроса 10

2.2.11 Запрос 11

Выбрать всех студентов, учащихся в классах с номером больше 5 и проживающих в Нью-Йорке.

```
SELECT * FROM main_scheme.student
WHERE class_id =
    ANY (SELECT id FROM main_scheme.class
         WHERE class_number > 5)
AND residential_address_id = ANY (SELECT id
         FROM main_scheme.residential_address
         WHERE city LIKE 'Нью%');
```

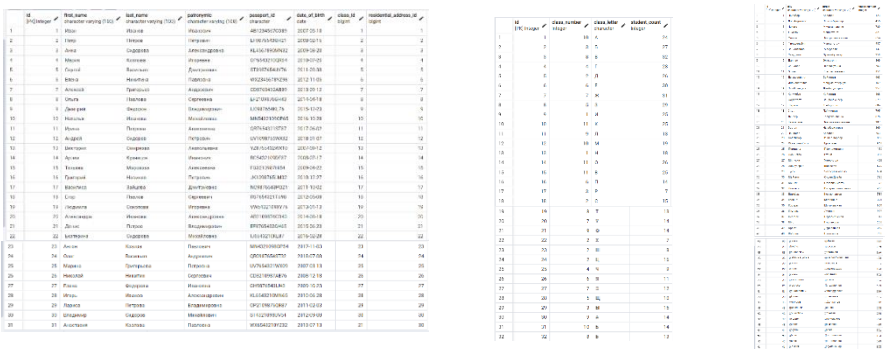


Рисунок 2.21 – Таблицы student, class и residential_address участвующая в запросе 11

Query Query History

```
1  SELECT * FROM main_scheme.student
2  WHERE class_id =
3      ANY(SELECT id
4          FROM main_scheme.class
5          WHERE class_number > 5)
6  AND residential_address_id =
7      ANY (SELECT id
8           FROM main_scheme.residential_address
9           WHERE city LIKE 'Нью%');
```

Data Output Messages Notifications

	id [PK] integer	first_name character varying (100)	last_name character varying (100)	middle_name character varying (100)	passport_id character	date_of_birth date	class_id bigint	residential_address_id bigint
1	1	Иван	Иванов	Иванович	AB1234567CD89	2007-05-10	1	1

Рисунок 2.22 – Результат выполнения запроса 11

2.2.12 Запрос 12

Найти самого старого сотрудника, вывести его профессию, город проживания и идентификационный номер паспорта.

```
SELECT
    employee.passport_id,
    residential_address.city,
    job_position.name AS profession
FROM main_scheme.employee
INNER JOIN main_scheme.residential_address
    ON residential_address.id =
        employee.residential_address_id
INNER JOIN main_scheme.employee_position
    ON employee_position.employee_id = employee.id
INNER JOIN main_scheme.job_position
    ON job_position.id =
        employee_position.job_position_id
WHERE date_of_birth <= ALL (SELECT date_of_birth
        FROM main_scheme.employee);
```

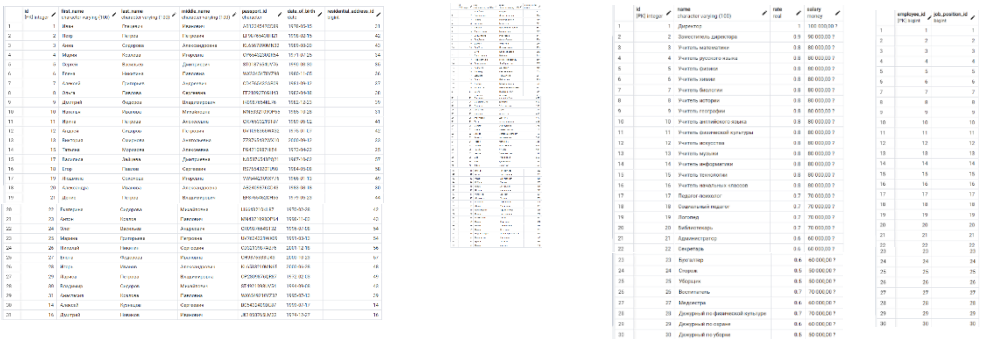


Рисунок 2.23 – Таблицы employee, residential_address, employee_position и job_position, участвующие в запросе 12

Query Query History

```
1  SELECT
2      employee.passport_id,
3      residential_address.city,
4      job_position.name AS profession
5  FROM main_scheme.employee
6  INNER JOIN main_scheme.residential_address
7  ON residential_address.id = employee.residential_address_id
8  INNER JOIN main_scheme.employee_position
9  ON employee_position.employee_id = employee.id
10 INNER JOIN main_scheme.job_position
11 ON job_position.id = employee_position.job_position_id
12 WHERE date_of_birth <= ALL (SELECT date_of_birth FROM main_scheme.employee);
13
```

Data Output Messages Notifications

	passport_id character	city character varying (100)	profession character varying (100)
1	VW5442109XY76	Фаниполь	Логопед

Рисунок 2.24 – Результат выполнения запроса 12

2.2.13 Запрос 13

Вывести фамилию учителя и его предмет, имеющего должность, в названии предмета которого есть слово “язык”.

```
SELECT
    employee.last_name,
    subject.name
FROM main_scheme.knowledge_of_subject
RIGHT OUTER
    JOIN main_scheme.employee
    ON employee.id =
    knowledge_of_subject.employee_id
RIGHT OUTER
    JOIN main_scheme.subject
    ON subject.id =
    knowledge_of_subject.subject_id
WHERE EXISTS
    (SELECT *
     FROM main_scheme.employee_position
     WHERE employee_position.employee_id =
        knowledge_of_subject.employee_id)
AND subject.name
LIKE '%язык%';
```

	id [PK] integer	name character varying (100)	hours integer	classroom integer
1	1	Математика	60	101
2	2	Физика	45	102
3	3	История	30	103
4	4	Литература	45	104
5	5	Биология	60	105
6	6	Химия	45	106
7	7	География	30	107
8	8	Английский язык	60	108
9	9	Французский язык	45	109
10	10	Немецкий язык	30	110
11	11	Испанский язык	45	111
12	12	Информатика	60	112
13	13	Физкультура	30	113
14	14	Музыка	45	114
15	15	ИЗО	60	115
16	16	Трудовое обучение	30	116
17	17	Обществознание	45	117
18	18	Право	60	118
19	19	Экономика	30	119
20	20	Психология	45	120
21	21	Философия	60	121
22	22	Религиоведение	30	122
23	23	Хореография	45	123
24	24	Театральное искусство	60	124
25	25	Основы медицины	30	125
26	26	Дизайн	45	126
27	27	Астрономия	60	127
28	28	Геология	30	128
29	29	Архитектура	45	129
30	30	Лингвистика	60	130

	id [PK] integer	first_name character varying (100)	last_name character varying (100)	middle_name character varying (100)	passport_id character	date_of_birth date	residential_address_id bigint
1	1	Иван	Гонцович	Иванович	AB1234567089	1978-05-15	31
2	2	Петр	Петров	Петрович	FF987654MH01	1988-07-15	42
3	3	Анна	Сидорова	Александровна	KL6567890NN32	1989-09-20	43
4	4	Мария	Козлова	Игоревна	QF5432100R54	1971-07-25	34
5	5	Сергей	Васильев	Дмитриевич	ST0187654UV76	1993-05-06	35
6	6	Елена	Новикова	Павловна	WX7345478VT98	1983-11-05	36
7	7	Алексей	Григорьев	Андреевич	GH7654102AB89	1981-09-12	37
8	8	Ольга	Павлова	Сергеевна	CD7109376GH43	1982-04-18	38
9	9	Дмитрий	Федоров	Владимирович	IK0987654KL76	1982-10-05	39
10	10	Наталья	Иванова	Михайловна	MN5332100PR65	1985-10-08	31
11	11	Ирина	Петрова	Алексеевна	QR7653021ST87	1989-05-02	41
12	12	Андрей	Сидоров	Петрович	UV1894365WX32	1975-01-07	42
13	13	Виктория	Смирнова	Анатольевна	ZZ8765432WX10	2003-09-12	33
14	15	Татьяна	Морозова	Алексеевна	FG4710987HI54	1973-04-22	35
15	17	Василиса	Зайцева	Дмитриевна	WX8976543RT021	1987-10-02	37
16	18	Егор	Павлов	Сергеевич	RS7654321YU90	1989-05-08	38
17	19	Людмила	Соколова	Игоревна	VW5447189XY76	1966-01-13	49
18	20	Александра	Иванова	Александровна	AB2409876CD43	1983-05-16	30
19	21	Денис	Петров	Владимирович	LH9785162GH55	1979-05-23	44
20	22	Екатерина	Сидорова	Михайловна	UM6152210KL87	1970-02-28	42
21	23	Антон	Козлов	Павлович	MN4321060DF64	1999-11-03	43
22	24	Олег	Васильев	Андреевич	QR9876543TS2	1995-07-08	54
23	25	Марина	Григорьева	Петровна	UV7439127KN09	1991-03-12	64
24	26	Николай	Новиков	Сергеевич	CD2129876MN76	2001-12-18	55
25	27	Елена	Федорова	Ивановна	GH9876533UH3	2003-10-23	57
26	28	Игорь	Иванов	Александрович	KL6532100NN65	2003-06-08	48
27	29	Лариса	Петрова	Владимировна	OP2309876GR87	1972-02-03	49
28	30	Владимир	Сидоров	Михайлович	SU1921989VX4	1991-09-08	43
29	31	Анастасия	Козлова	Павловна	WX5491210YZ32	1995-07-13	39
30	14	Алексей	Кузнецов	Сергеевич	DC5432400EC07	1999-07-17	14
31	16	Дмитрий	Новиков	Иванович	JK7098765MN2	1974-12-27	16

	employee_id [PK] bigint	job_position_id bigint
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30

	employee_id [PK] bigint	subject_id bigint
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	7	7
8	8	8
9	9	9
10	10	10
11	11	11
12	12	12
13	13	13
14	14	14
15	15	15
16	16	16
17	17	17
18	18	18
19	19	19
20	20	20
21	21	21
22	22	22
23	23	23
24	24	24
25	25	25
26	26	26
27	27	27
28	28	28
29	29	29
30	30	30

Рисунок 2.25 – Таблицы subject, employee, employee_position и knowledge_of_subject, участвующие в запросе 13

Результат выполнения запроса представлен на рисунке 2.26.

Query Query History

```
1  SELECT
2      employee.last_name,
3      subject.name
4  FROM main_scheme.knowledge_of_subject
5  RIGHT OUTER JOIN main_scheme.employee
6  ON employee.id = knowledge_of_subject.employee_id
7  RIGHT OUTER JOIN main_scheme.subject
8  ON subject.id = knowledge_of_subject.subject_id
9  WHERE EXISTS
10     (SELECT *
11      FROM main_scheme.employee_position
12      WHERE employee_position.employee_id = knowledge_of_subject.employee_id)
13 AND subject.name LIKE '%язык%';
14
```

Data Output Messages Notifications

	last_name character varying (100)	name character varying (100)
1	Капуста	Английский язык
2	Федоров	Французский язык
3	Иванова	Немецкий язык
4	Петрова	Испанский язык

Рисунок 2.26 – Результат выполнения запроса 13

2.2.14 Запрос 14

Вывести фамилии всех студентов, их город, улицу и номер дома, которые живут в Нью-Йорке, и их оценку.

```
SELECT
    student.last_name,
    residential_address.city,
    residential_address.street,
```

```
    residential_address.house_number,
    gradebook.grade
FROM main_scheme.student
FULL OUTER JOIN main_scheme.residential_address
ON student.residential_address_id =
    residential_address.id
FULL OUTER JOIN main_scheme.class
ON class.id = student.class_id
FULL OUTER JOIN main_scheme.gradebook
ON gradebook.student_id = student.id
            WHERE city = 'Нью-Йорк';
```

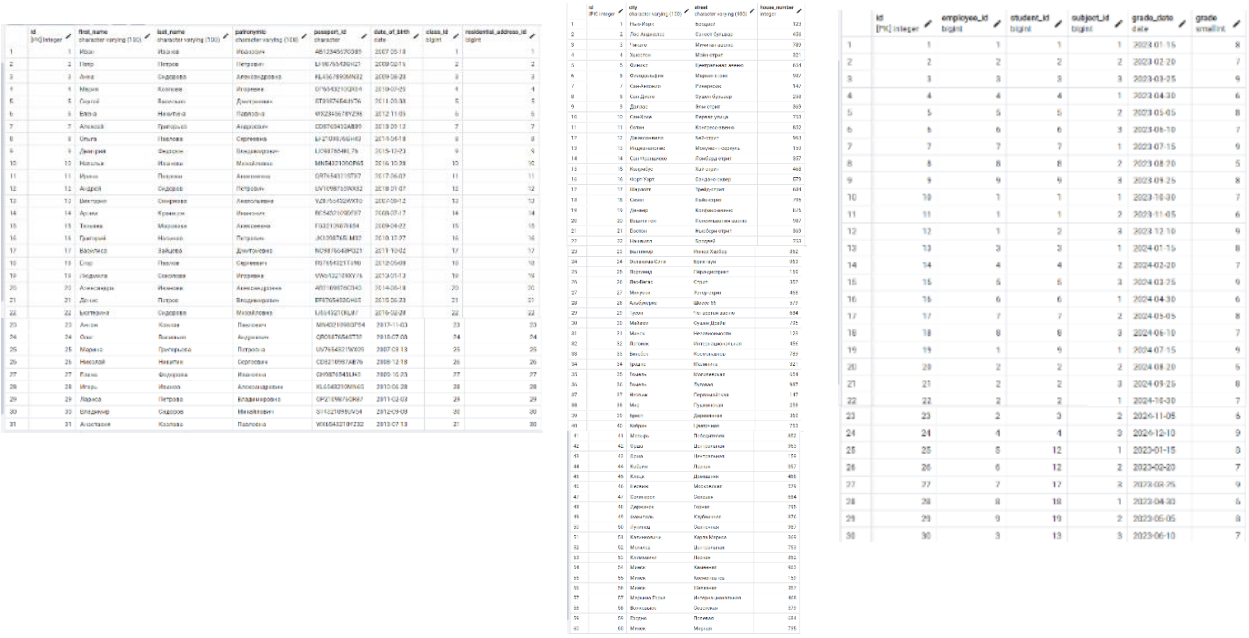
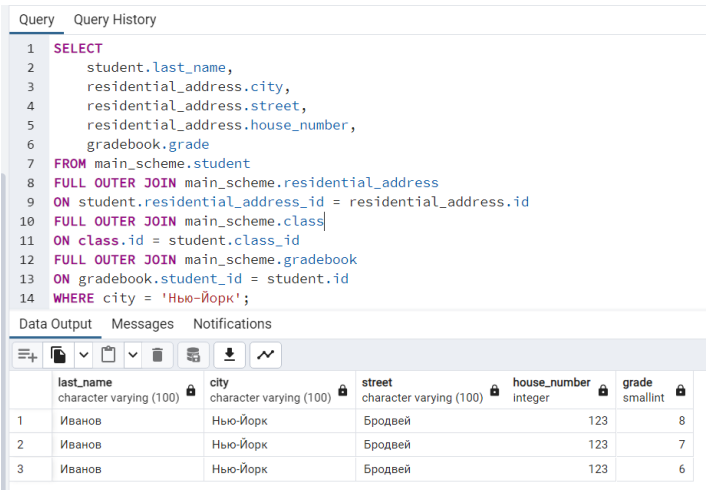


Рисунок 2.27 – Таблицы student, residential_address и gradebook, участвующие в запросе 14



Query Query History

```
SELECT
    student.last_name,
    residential_address.city,
    residential_address.street,
    residential_address.house_number,
    gradebook.grade
FROM main_scheme.student
FULL OUTER JOIN main_scheme.residential_address
ON student.residential_address_id = residential_address.id
FULL OUTER JOIN main_scheme.class
ON class.id = student.class_id
FULL OUTER JOIN main_scheme.gradebook
ON gradebook.student_id = student.id
WHERE city = 'Нью-Йорк';
```

Data Output Messages Notifications

	last_name character varying (100)	city character varying (100)	street character varying (100)	house_number integer	grade smallint
1	Иванов	Нью-Йорк	Бродвей	123	8
2	Иванов	Нью-Йорк	Бродвей	123	7
3	Иванов	Нью-Йорк	Бродвей	123	6

Рисунок 2.28 – Результат выполнения запроса 14

2.2.15 Запрос 15

Вывести название должности, ФИО сотрудника, оклад которого составляет 70000 условных денежных единиц.

```
SELECT
    job_position.name,
    job_position.salary,
    employee.last_name,
    employee.first_name,
    employee.middle_name
FROM main_scheme.employee_position
FULL OUTER JOIN main_scheme.job_position
ON employee_position.job_position_id =
    job_position.id
INNER JOIN main_scheme.employee
ON employee.id = employee_position.employee_id
WHERE salary = '70000';
```

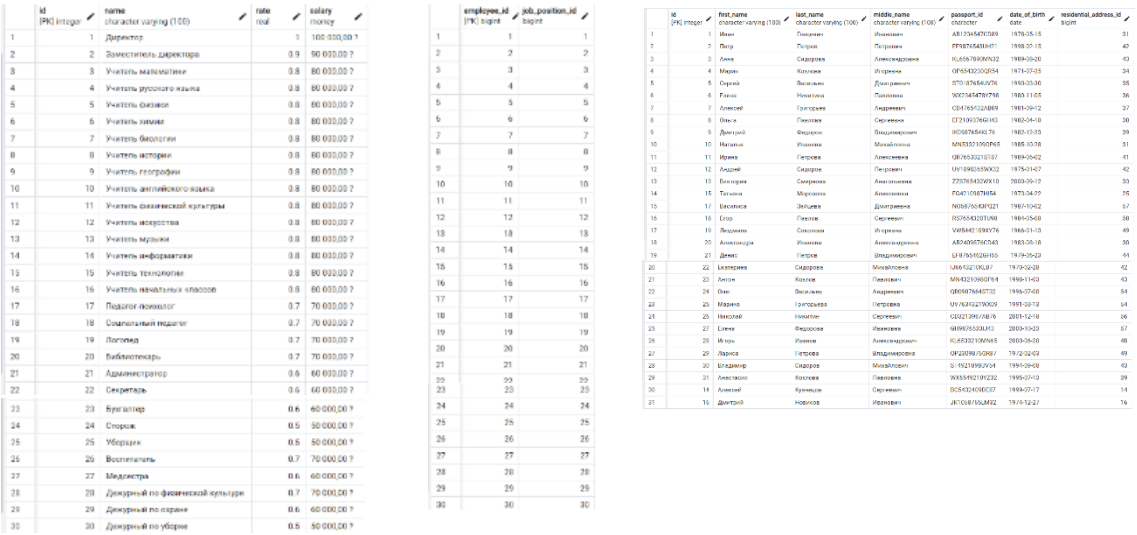


Рисунок 2.29 – Таблицы job_position, employee_position и employee, участвующие в запросе 15

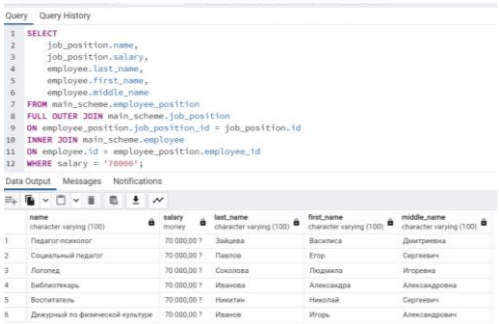


Рисунок 2.30 – Результат выполнения запроса 15

4 ВЫВОД

В ходе лабораторной работе было выполнено создание запросов на выборку данных на языке SQL с использованием подзапросов, агрегатных функций, а также группировки данных (предложение `GROUP BY` оператора `SELECT`) и операций над множествами (`UNION`, `INTERSECT`, `EXCEPT`).