ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ПЕТРА ВЕЛИКОГО**
Институт компьютерных наук и технологий
Системы искусственного интеллекта и суперкомпьютерные технологии
Направление 02.03.01 Математика и Компьютерные науки

Отчёт по дисциплине «Теоретические основы баз данных»

Курсовая работа

«База данных для управления онлайн школой»

Студент группы 5130201/00101: _____ Кулыгин Егор Александрович

Преподаватель: _____ Попов Сергей Геннадьевич

«__» _____ 2023г.

Санкт-Петербург – 2023

Содержание

1 Аналитика

1.1 Описание предметной области

Образование и подготовка к ЕГЭ (единый государственный экзамен) являются важными этапами в жизни молодых людей, которые влияют на их будущее образование и карьеру. Подготовка к ЕГЭ стала актуальной и востребованной областью, поскольку результаты этого экзамена влияют на поступление в высшие учебные заведения.
Понятия:

- **Единый государственный экзамен (ЕГЭ)**: Это система обязательных экзаменов, проводимых во многих странах, в том числе и в России, для оценки знаний учащихся и студентов, а также для определения их готовности к поступлению в высшие учебные заведения.
- **Онлайн-школа**: Образовательное учреждение, которое предоставляет обучение и подготовку в интернете, позволяя ученикам получать знания и навыки удаленно, через онлайн-платформы.
- **Вебинары**: Интерактивные онлайн-уроки, проводимые в режиме реального времени через интернет.
- **Старшие и младшие преподаватели**: Категории преподавателей в онлайн-школе, где старшие преподаватели выполняют роль наставников и администраторов, а младшие преподаватели обучают учеников на личных уроках.

Онлайн-школа подготовки к ЕГЭ "ОГО-ЕГЭ"предоставляет возможность ученикам получить качественную подготовку к экзаменам в удобной форме. Школа предлагает обширный спектр образовательных услуг и курсов, охватывая разнообразные предметные области, такие как математика, информатика, русский язык и многие другие.

Школа предоставляет услугу подготовки к ЕГЭ. Эта услуга включает в себя:

1. Проведение онлайн уроков в виде прямых эфиров с разбором всех задач экзамена.
2. Подготовку и проверку домашних заданий по пройденному материалу.
3. Проведение уроков с учеником лично для более гибкого подхода.

Старшие преподаватели занимаются ведением вебинаров и являются наставниками для младших. Они отвечают на вопросы младших преподавателей, проводят случайные проверки и оценивают качество проводимых младшими преподавателями занятий. По итогам вебинаров ученик может пройти тест и оценить качество усвоенного материала.

Младшие преподаватели занимаются проведением уроков лично с учениками и не проводят вебинары. Также они проверяют домашние задания учеников.

Время полной подготовки рассчитано на 9 месяцев. Начинается курс каждый понедельник 2 недели сентября и продолжается ровно до 2 недели мая. При этом школа предусматривает возможность начать обучение уже во время курса.

Создание базы данных играет ключевую роль в современной системе образования, обеспечивая улучшение качества обучения, управление ресурсами и развитие школы, а также повышая удовлетворенность учеников и преподавателей.

1.2 Выделение целей создания системы

Цели создания системы:

- Учет выполнения домашних заданий
- Управление преподавателями
- Планирование вебинаров
- Управление предметами
- Учет студентов
- Управление учебными материалами
- Анализ и отчетность
- Распределение младших преподавателей среди старших
- Распределение учеников среди младших преподавателей

1.3 Выделение сущностей и их атрибутов

Выделены следующие сущности и их атрибуты:

1. Ученики (`students`)
    - `Имя`
    - `Дата рождения`
    - `Электронная почта`
    - `Номер телефона`
2. Младшие преподаватели (`teachers`)
    - `Имя`
    - `Заработная плата`
    - `Электронная почта`
    - `Номер телефона`
    - `Преподаваемый этим учителем предмет`
3. Предметы (`subjects`)
    - `Название предмета`
4. Вебинары (`webinars`)
    - `Дата проведения вебинара`
    - `Тема вебинара`
5. Оценки за вебинары (`grades`)
    - `Студент, для которого выставлена оценка`
    - `Вебинар, на котором выставлена оценка`

- Выставленная оценка

6. Домашние задания (homework)
    - Предмет по которому задано домашнее задание
    - Название домашнего задания
    - Дата открытия домашнего задания то есть день, месяц и год, когда у учеников появится доступ к выполнению урока
    - Максимальное количество баллов
    - Крайний срок сдачи домашнего задания
7. Результаты домашних заданий (homeworkresults)
    - Студент, который сдал домашнее задание
    - Полученное студентом количество баллов
8. Уроки (lessons)
    - Преподаватель, который проводит урок
    - Студент, с которым проводится урок
    - Предмет, по которому ведется урок
    - День недели, когда проводится урок
    - Время начала урока, по умолчанию возможно с 8:00 утра до 18:00 вечера
9. Старшие преподаватели (seniorteachers)
    - Имя
    - Заработная плата
    - Электронная почта
    - Номер телефона
    - Преподаваемый этим учителем предмет

1.4 ER-Диаграмма

На Рис. 1 представлена ER-Диаграмма для построенной базы данных.

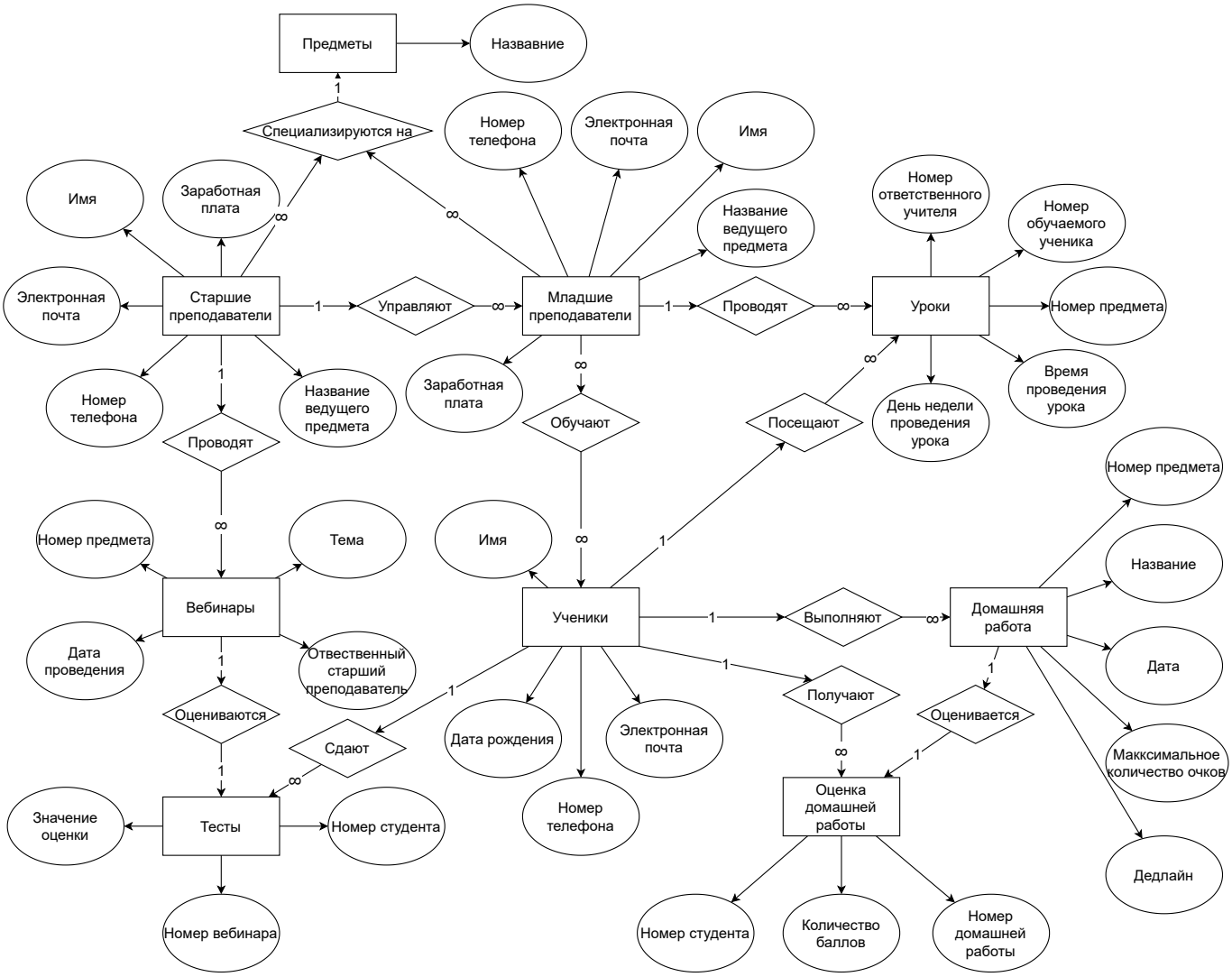

Рис. 1: ER-Диаграмма

1.4.1 Чтение ER-Диаграммы

Примеры чтения ER-Диаграммы:

- Старшие преподаватели управляют младшеми преподавателями
- Старшие преподаватели проводят вебинары, которые оцениваются тестами
- Ученики выполняют домашнюю работу и получают оценку домашней работы
- Младшие преподаватели обучают учеников, которые посещают уроки
- Младшие преподаватели специализируются на предметах

1.5 Схема объектов

На Рис. 2 представлена схема объектов для выбранной предметной области.

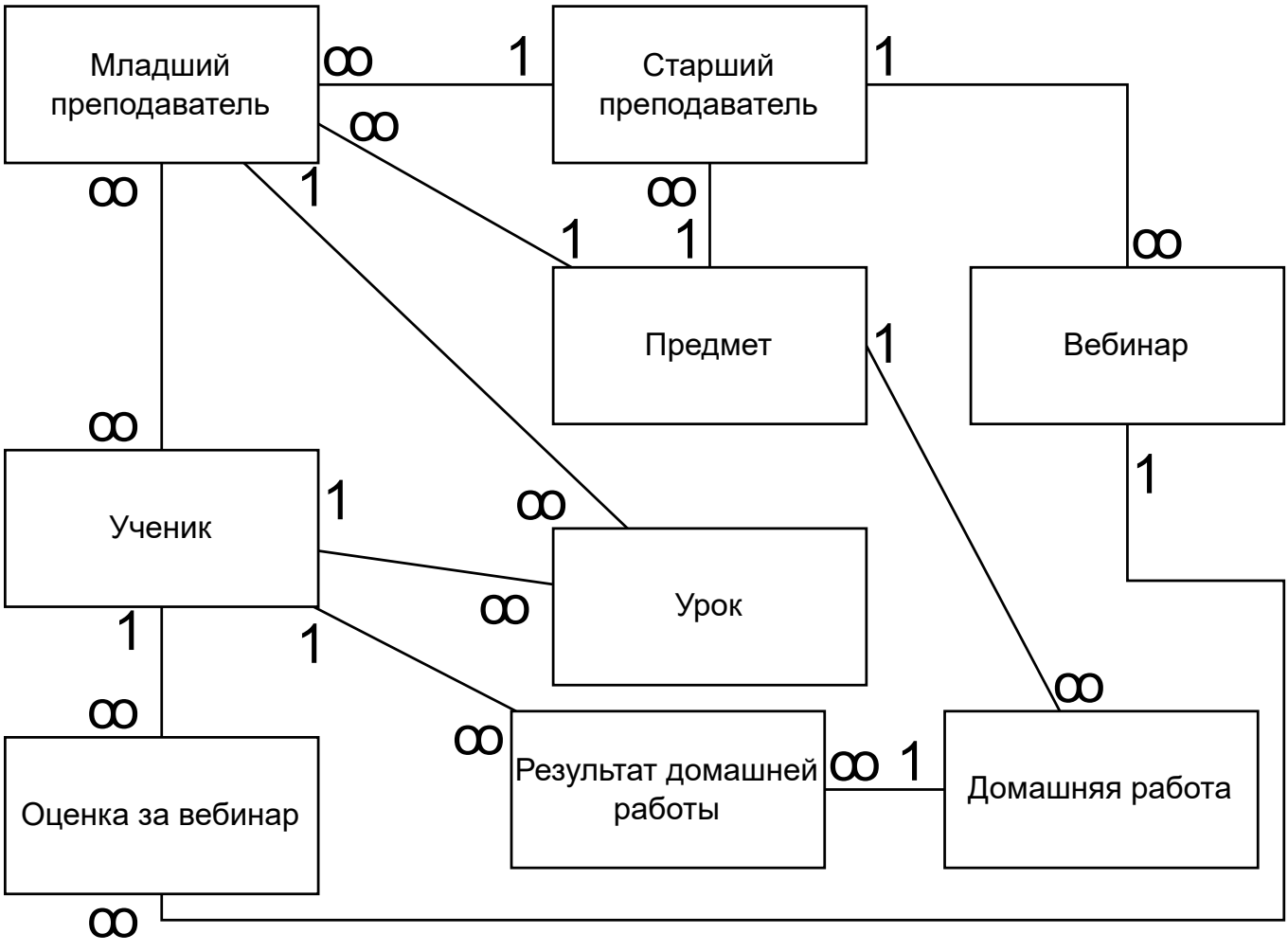

Рис. 2: Схема объектов

2 Проектирование

2.1 Схема базы данных

На Рис. 3 представлена схема базы данных на английском языке.

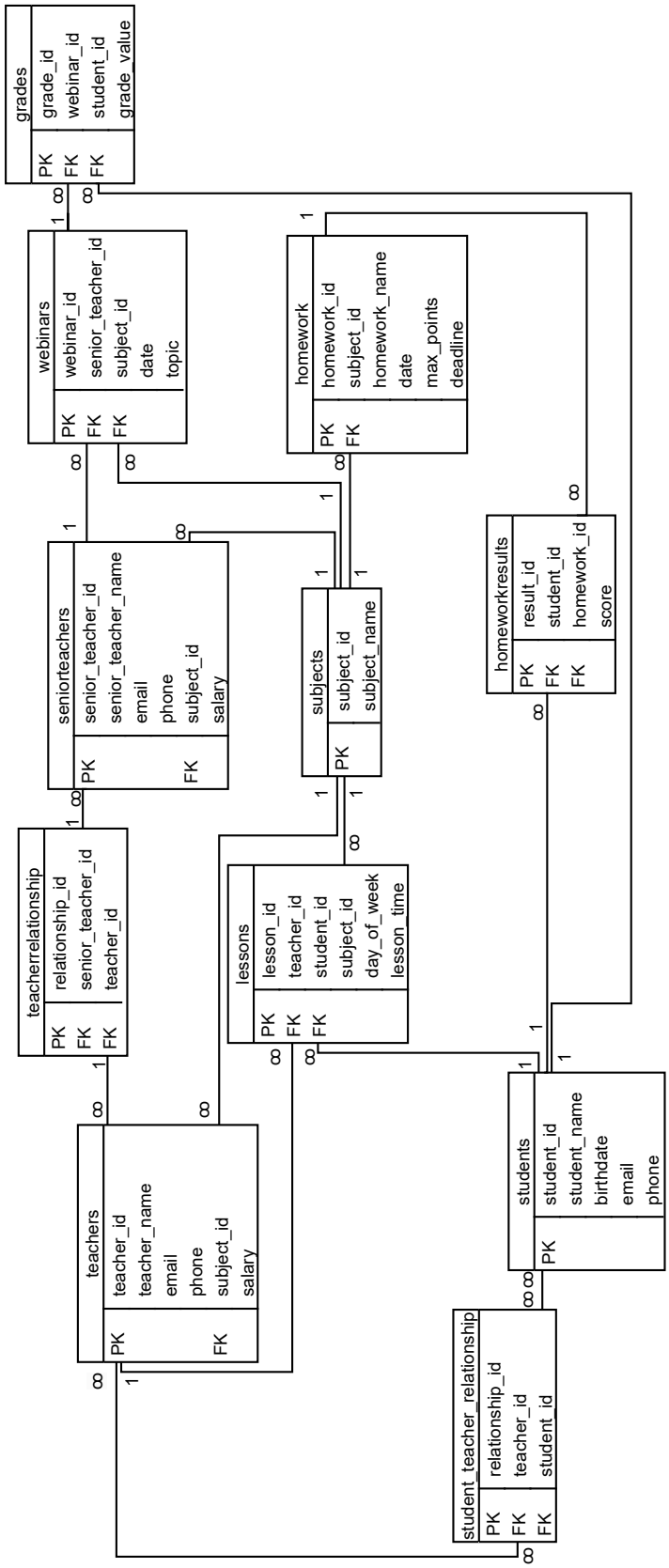

Рис. 3: Схема базы данных на английском языке

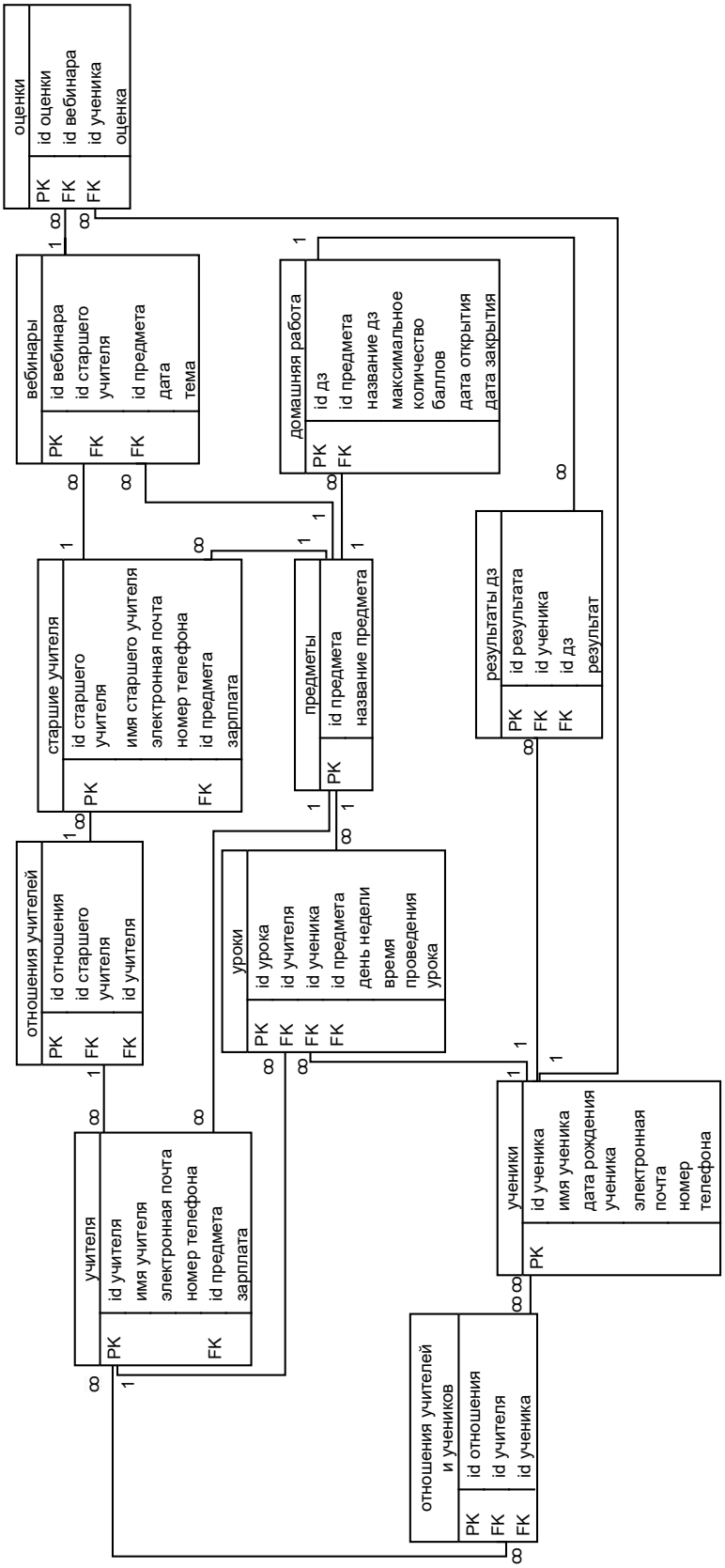

Рис. 4: Схема базы данных на русском языке

2.2 Таблицы базы данных

В Таблицах 1-11 представлены все атрибуты базы данных.

Название поля	**Тип данных**	**Тип ключа**	**Ссылка**	**Ограничения**
grade_id	INT	PK	-	NN AI
student_id	INT	FK	students(student_id)	NN
webinar_id	INT	FK	webinars(webinar_id)	NN
grade_value	TINYINT	-	-	NN UN

Таблица 1: Структура таблицы grades

Название поля	**Тип данных**	**Тип ключа**	**Ссылка**	**Ограничения**
homework_id	INT	PK	-	NN AI
subject_id	INT	FK	subjects(subject_id)	NN
homework_name	VARCHAR(255)	-	-	NN
date	DATE	-	-	NN
max_points	TINYINT	-	-	NN
deadline	DATE	-	-	NN

Таблица 2: Структура таблицы homework

Название поля	**Тип данных**	**Тип ключа**	**Ссылка**	**Ограничения**
result_id	INT	PK	-	NN AI
student_id	INT	FK	students(student_id)	NN
homework_id	INT	FK	homework(homework_id)	NN
score	TINYINT	-	-	NN UN

Таблица 3: Структура таблицы homeworkresults

Название поля	**Тип данных**	**Тип ключа**	**Ссылка**	**Ограничения**
lesson_id	INT	PK	-	NN AI
teacher_id	INT	FK	teachers(teacher_id)	NN
student_id	INT	FK	students(student_id)	NN
subject_id	INT	FK	subjects(subject_id)	NN
day_of_week	VARCHAR(15)	-	-	NN
lesson_time	TIME	-	-	NN

Таблица 4: Структура таблицы lessons

Название поля	**Тип данных**	**Тип ключа**	**Ссылка**	**Ограничения**
senior_teacher_id	INT	PK	-	NN AI
senior_teacher_name	VARCHAR(255)	-	-	NN
email	VARCHAR(255)	-	-	NN
phone	VARCHAR(15)	-	-	NN
subject_id	INT	FK	subjects(subject_id)	NN
salary	INT	-	-	NN UN

Таблица 5: Структура таблицы seniorteachers

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
relationship_id	INT	PK	-	NN AI
student_id	INT	FK	students(student_id)	NN
teacher_id	INT	FK	teachers(teacher_id)	NN

Таблица 6: Структура таблицы student_teacher_relationship

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
student_id	INT	PK	-	NN AI
student_name	VARCHAR(255)	-	-	NN
birthdate	DATE	-	-	NN
email	VARCHAR(255)	-	-	NN
phone	VARCHAR(15)	-	-	NN

Таблица 7: Структура таблицы students

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
subject_id	INT	PK	-	NN AI
subject_name	VARCHAR(30)	-	-	NN

Таблица 8: Структура таблицы subjects

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
relationship_id	INT	PK	-	NN AI
senior_teacher_id	INT	FK	seniorteachers(senior_teacher_id)	NN
junior_teacher_id	INT	FK	teachers(teacher_id)	NN

Таблица 9: Структура таблицы teacherrelationship

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
teacher_id	INT	PK	-	NN AI
teacher_name	VARCHAR(255)	-	-	NN
email	VARCHAR(255)	-	-	NN
phone	VARCHAR(15)	-	-	NN
subject_id	INT	FK	subjects(subject_id)	NN
salary	INT	-	-	NN UN

Таблица 10: Структура таблицы teachers

Название поля	Тип данных	Тип ключа	Ссылка	Ограничения
webinar_id	INT	PK	-	NN AI
subject_id	INT	FK	subjects(subject_id)	NN
date	DATE	-	-	NN
topic	VARCHAR(100)	-	-	NN
senior_teacher_id	INT	FK	seniorteacher(senior_teacher_id)	NN

Таблица 11: Структура таблицы webinars

2.3 Обоснование выбранных типов данных

- **INT**: Для всех первичных и внешних ключей (`grade_id`, `student_id`, `webinar_id`, `subject_id`, `homework_id`, `homeworkresults_id`, `lesson_id`, `senior_teacher_id`, `junior_teacher_id`, `teacher_id`, `relationship_id`, `subject_id`) выбран тип данных `INT`. Это обеспечивает целочисленное хранение значений до $2^{31}-1$. Также, добавлено ограничение AI, которое предоставляет автоматическое увеличение значений, что удобно для уникальных идентификаторов и связей между таблицами. Помимо этого, большинство полей имеет флаг NOT NULL для обязательного заполнения данных значений.

- **TINYINT**: Для полей, хранящих оценки (`grade_value`) и максимальное количество баллов (`max_points`), выбран тип `TINYINT`, так как оценки и баллы имеют ограниченный и меньший диапазон значений.

- **VARCHAR**: Для текстовых полей, таких как названия (`homework_name`, `senior_teacher_name`, `student_name`, `email`), выбран тип `VARCHAR(255)`. Это позволяет хранить текстовые данные переменной длины до 255 символов, что достаточно для описания объектов. При этом в для поля номера телефона выбран тип `VARCHAR(15)`, чтобы хранить номера телефонов в формате +7(XXX)XXX-XX-XX и других схожих форматах. Также, меньшую длину `VARCHAR(9)` имеет день недели, так как максимальная длина это Wednesday (9 букв).

- **DATE**: Для хранения дат (`birthdate`, `date`, `deadline`), выбран тип `DATE`, который хранит дату в формате YYYY-MM-DD.

- **TIME**: Для хранения времени (`lesson_time`), выбран тип `TIME`, который позволяет хранить время в формате часы:минуты:секунды. Секунды являются лишним показателем и не заполняются, но включены в сам тип `TIME`.

2.4 Генерация записей

Для генерации записей в таблице мной было реализовано 9 python скриптов, которые позволяют заполнять таблицы данными. Они получают на вход аргумент, обозначающий какое количество записей необходимо добавить, что позволяет запускать их через консоль для быстрого применения.

Помимо этого, также были реализованы вспомогательные скрипты которые организуют связи между уже заполненными таблицами. Они нужны для генерации таблиц формата many to many.

Для генерации записей в данных скриптах мной использовались такие команды как insert, delete, update, set и многие другие. Также, была сгенерирована база имён и фамилий для заполнения соответствующих полей путём их случайного сочетания.

Частично ручным образом заполнена таблица webinars. Множество тем (поле topic) было взято с сайта РЕШУ ЕГЭ, оно биективно множеству задач на ЕГЭ по каждому предмету. Далее использовался скрипт findTeacherForWebinar.py, который находит ведущих из пула старших преподавателей.

Полностью ручным образом заполнена база предметов (subjects). Мной были выбраны самые частые для сдачи предметы ЕГЭ.

Table Name	Row Count
grades	66440
homework	250
homeworkresults	200030
lessons	69957
seniorteachers	15124
student_teacher_relationship	404874
students	133995
subjects	5
teacherrelationship	120516
teachers	119570
webinars	153
summary	1130914

Таблица 12: Общее число строк в каждой из таблиц

3 Составление запросов

3.1 Запрос №1

Формулировка запроса

Найти всех старших учителей, которые поставили оценку 1 ученику ID = 183441

Код запроса

```
SELECT t.senior_teacher_name,g.student_id,g.grade_value
FROM grades AS g
JOIN webinars AS w ON g.webinar_id = w.webinar_id
JOIN seniorteachers AS t ON w.senior_teacher_id = t.senior_teacher_id
WHERE g.student_id = 183441
AND g.grade_value = 1;
```

Результат запроса

Запрос выполнился за 16 миллисекунд. В результирующей таблице получилось 3 строки.

senior_teacher_name	student_id	grade_value
Артем Дорофеев	183441	1
Викентий Семенов	183441	1
Денис Королёв	183441	1

Рис. 5: Результат первого запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	g	NULL	ref	fk_student_id,fk_webinar_id	fk_student_id	4	const	4	10.00	Using where
1	SIMPLE	w	NULL	eq_ref	PRIMARY,idx_webinar_id	PRIMARY	4	ege_online_school.g.webinar_id	1	100.00	NULL
1	SIMPLE	t	NULL	eq_ref	PRIMARY	PRIMARY	4	ege_online_school.w.senior_teacher_id	1	100.00	NULL

Рис. 6: Результат EXPLAIN первого запроса

Объяснение запроса

В данном запросе выполняется объединение таблиц grades, webinars и seniorteachers по соответствующим идентификаторам. Запрос выбирает только строки, где student_id равен 183441 и grade_value равен 1..

3.2 Запрос №2

Формулировка запроса

Посчитать число учеников, получших по предмету ID = 2 суммарно 20 баллов за домашние задания .

Код запроса

```
SELECT subq.subject_id as subject,
count(*) as Number_of_Students,subq.total_score
FROM(
SELECT subject_id, SUM(sr.score) AS total_score, st.student_id
FROM homeworkresults AS sr
JOIN homework AS h ON sr.homework_id = h.homework_id
JOIN students AS st ON st.student_id = sr.student_id
WHERE subject_id = 2
GROUP BY subject_id, st.student_id
HAVING SUM(sr.score) = 20) AS subq
group by subq.subject_id,subq.total_score;
```

Результат запроса

Запрос выполнился за 625 миллисекунд. В результирующей таблице получилась 1 строка.

	subject	Number_of_Students	total_score
▶	2	732	20

Рис. 7: Результат второго запроса

	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	41328	100.00	Using temporary
	2	DERIVED	h	NULL	ref	PRIMARY,homework_id_UNIQUE,idx_homework_id,homework_ibfk_1	homework_ibfk_1	4	const	50	100.00	Using index; Using temporary
	2	DERIVED	sr	NULL	ref	homeworkresults_ibfk_1,homeworkresults_ibfk_2	homeworkresults_ibfk_2	4	ege_online_school.h.homework_id	826	100.00	NULL
	2	DERIVED	st	NULL	eq_ref	PRIMARY	PRIMARY	4	ege_online_school.sr.student_id	1	100.00	Using index

Рис. 8: Результат EXPLAIN второго запроса

Объяснение запроса

В данном SQL-запросе выполняется выборка данных из таблиц homeworkresults, homework и students для предмета с идентификатором subject_id равным 2. Затем результаты группируются по subject_id и student_id, и выбираются только те записи, где сумма оценок (score) равна 20. Наконец, результаты агрегируются снова по subject_id и total_score, подсчитывая количество студентов и общий балл.

3.3 Запрос №3.1

Формулировка запроса

Найти учеников с максимальным количеством посещений вебинаров.

Код запроса

```
SELECT student_id, COUNT(grade_id) AS num_grades
FROM grades
GROUP BY student_id
HAVING num_grades = (
SELECT  COUNT(grade_id) AS num_grades
```

```
FROM grades
GROUP BY student_id
order by  COUNT(grade_id) desc
limit 1
);
```

Результат запроса

Запрос выполнился за 47 миллисекунды. В результирующей таблице получилось 28 строк, на скриншоте приведены первые 5 строк.

student_id	num_grades
183366	6
184283	6
184641	6
188832	6
188933	6

Рис. 9: Результат 3.1 запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	grades	NULL	index	fk_student_id	fk_student_id	4	NULL	73264	100.00	Using index
2	SUBQUERY	grades	NULL	index	fk_student_id	fk_student_id	4	NULL	73264	100.00	Using index; Using temporary; Using filesort

Рис. 10: Результат EXPLAIN 3.1 запроса

Объяснение запроса

Внешний запрос выбирает student_id и подсчитывает количество grade_id для каждого студента из таблицы grades. Затем происходит группировка результатов по student_id с использованием оператора GROUP BY. Затем используется оператор HAVING для фильтрации результатов, оставляя только те строки, в которых количество оценок (num_grades) равно максимальному значению количества оценок в подзапросе. В подзапросе (subquery) выполняются аналогичные операции. Он снова выбирает student_id и подсчитывает количество grade_id для каждого студента, группируя результаты по student_id. В этом подзапросе также выбирается максимальное количество оценок с помощью оператора MAX(subquery.num_grades).

3.4 Запрос №3.2

Формулировка запроса

Найти учеников с минимальным количеством посещений вебинаров.

Код запроса

```
SELECT student_id, COUNT(grade_id) AS num_grades
FROM grades
GROUP BY student_id
HAVING num_grades = (
SELECT  COUNT(grade_id) AS num_grades
FROM grades
```

```
GROUP BY student_id
order by  COUNT(grade_id)
limit 1
);
```

Результат запроса

Запрос выполнился за 31 миллисекунд. В результирующей таблице получилось 43090 строк.

student_id	num_grades
181853	1
181854	1
181855	1

Рис. 11: Результат 3.2 запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	grades	NULL	index	fk_student_id	fk_student_id	4	NULL	73264	100.00	Using index
2	SUBQUERY	grades	NULL	index	fk_student_id	fk_student_id	4	NULL	73264	100.00	Using index; Using temporary; Using filesort

Рис. 12: Результат EXPLAIN 3.2 запроса

Объяснение запроса

Аналогично запросу 3.1 но с минимум.

3.5 Запрос №4

Формулировка запроса

Найти учителей, которые провели больше уроков по предмету ID =1, чем учитель ID = 3.

Код запроса

```
    SELECT
    tch_outer.teacher_id,
    COUNT(ls_outer.lesson_id) AS cnt,
        (
        SELECT subject_id
        FROM teachers AS tch_inner
        WHERE tch_inner.teacher_id = 3
    ) AS inner_subject_id
FROM teachers AS tch_outer
JOIN lessons AS ls_outer ON ls_outer.teacher_id = tch_outer.teacher_id
GROUP BY tch_outer.teacher_id, tch_outer.subject_id
HAVING cnt >     (
        SELECT COUNT(ls_inner.lesson_id)
        FROM teachers AS tch_inner
        JOIN lessons AS ls_inner ON ls_inner.teacher_id = tch_inner.teacher_id
        WHERE tch_inner.teacher_id = 3
```

```
);
```

Результат запроса

Запрос выполнился за 500 миллисекунд. В результирующей таблице получилось 27969 строк, на скриншоте приведены первые 5 строк.

teacher_id	cnt	inner_subject_id
5	3	1
7	3	1
10	3	1
17	3	1
22	3	1

Рис. 13: Результат пятого запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	tch_outer	NULL	index	PRIMARY,teachers_i...	teachers_ibfk_1	4	NULL	119570	100.00	Using index; Using temporary
1	PRIMARY	ls_outer	NULL	ref	teacher_id	teacher_id	4	ege_online_school.tch_outer.teacher_id	1	100.00	Using index
3	SUBQUERY	tch_inner	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	Using index
3	SUBQUERY	ls_inner	NULL	ref	teacher_id	teacher_id	4	const	2	100.00	Using index
2	SUBQUERY	tch_inner	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL

Рис. 14: Результат EXPLAIN пятого запроса

Объяснение запроса

Во внешнем запросе выбираются учителя (teacher_id) и подсчитывается количество уроков (lesson_id), которые им принадлежат. Это выполняется с помощью соединения (JOIN) таблиц teachers и lessons, где teacher_id в обеих таблицах совпадает. Затем выполняется подзапрос (subquery), который выбирает subject_id из таблицы teachers для учителя с teacher_id равным 3. Результат внешнего запроса группируется по teacher_id и subject_id учителей из таблицы teachers. Оператор HAVING фильтрует результаты, оставляя только те строки, где количество уроков (cnt) больше, чем количество уроков, принадлежащих учителю с teacher_id равным 3. Для этого используется подзапрос в HAVING, который подсчитывает количество уроков для учителя с teacher_id равным 3.

3.6 Запрос №5

Формулировка запроса

Найти учителей с одинаковым числом уроков.

Код запроса

```
    SELECT num_lessons AS "Число уроков", COUNT(*) AS "Число учителей"
FROM (
    SELECT teacher_id, COUNT(*) AS num_lessons
    FROM lessons
    GROUP BY teacher_id
    UNION ALL
    SELECT teacher_id, 0 AS num_lessons
    FROM teachers
```

```
    WHERE teacher_id NOT IN (SELECT DISTINCT teacher_id FROM lessons)
) AS subquery
GROUP BY num_lessons
order by num_lessons asc;
```

Результат запроса

Запрос выполнился за 203 миллисекунд. В результирующей таблице получилось 11 строк.

Число уроков	Число учителей
0	22589
1	38020
2	31534
3	17324
4	7386
5	2402
6	661
7	157
8	33
9	3
10	3

Рис. 15: Результат пятого запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	303060	100.00	Using temporary; Using filesort
2	DERIVED	lessons	NULL	index	teacher_id	teacher_id	4	NULL	183490	100.00	Using index
3	UNION	teachers	NULL	index	NULL	teachers_ibfk_1	4	NULL	119570	100.00	Using index
3	UNION	<subquery4>	NULL	eq_ref	<auto_distinct_key>	<auto_distinct_key>	5	ege_online_school.teachers.teacher_id	1	100.00	Using where; Not exists
4	MATERIALIZED	lessons	NULL	index	teacher_id	teacher_id	4	NULL	183490	100.00	Using index

Рис. 16: Результат EXPLAIN пятого запроса

Гистограмма, построенная по запросу

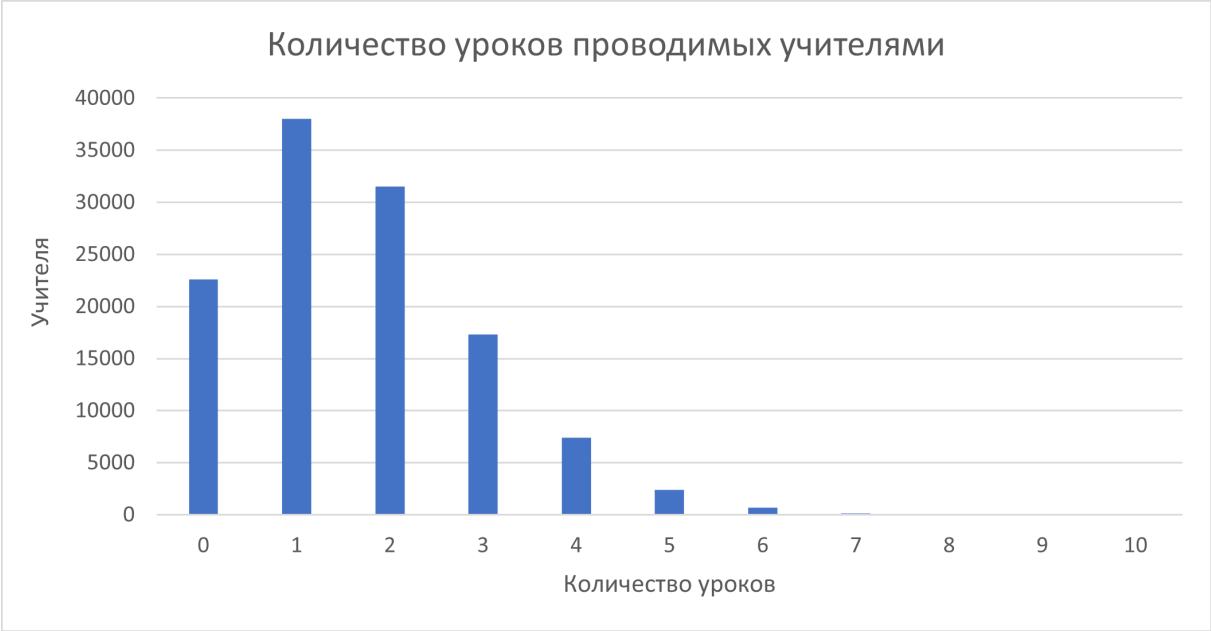

Рис. 17: Гистограмма пятого запроса

Объяснение запроса

Сначала создается подзапрос (subquery), который объединяет два набора данных с использованием оператора UNION ALL:

Первая часть подзапроса выбирает учителей из таблицы lessons и подсчитывает количество уроков (num_lessons) для каждого учителя, используя оператор GROUP BY. Вторая часть подзапроса выбирает учителей из таблицы teachers, которые не имеют записей в таблице lessons, и устанавливает num_lessons равным 0. Затем внешний запрос выбирает num_lessons (число уроков) и подсчитывает количество учителей (COUNT(*)) для каждой категории числа уроков.

Результаты группируются по num_lessons.

Результаты сортируются в порядке возрастания num_lessons с использованием оператора ORDER BY.

3.7 Запрос №6

Формулировка запроса

Найти преподавателей, которые не вели уроков у ученика Карина Шестякова ID = 208269.

Код запроса

```
    SELECT t.teacher_name as 'Преподаватель', s.student_name as 'Не ведет'
FROM teachers t
LEFT JOIN (
    SELECT DISTINCT t.teacher_id
    FROM teachers t
    JOIN lessons l ON t.teacher_id = l.teacher_id AND l.student_id = 208269
) AS subquery ON t.teacher_id = subquery.teacher_id
```

```
LEFT JOIN students s ON s.student_id = 208269
WHERE subquery.teacher_id IS NULL;
```

Результат запроса

Запрос выполнился за 16 миллисекунд. В результирующей таблице получилось 120109 строк, на скриншоте приведены первые 5 строк.

Преподаватель	Не ведет
Максим Осипов	Карина Шестакова
Виталий Лебедев	Карина Шестакова
Константин Петухов	Карина Шестакова
Валерий Рябов	Карина Шестакова
Маргарита Щербакова	Карина Шестакова

Рис. 18: Результат шестого запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	PRIMARY	t	NULL	ALL	NULL	NULL	NULL	NULL	119570	100.00	NULL
1	PRIMARY	<derived2>	NULL	ALL	NULL	NULL	NULL	NULL	3	33.33	Using where; Not exists; Using join buffer (hash...
1	PRIMARY	s	NULL	const	PRIMARY	PRIMARY	4	const	1	100.00	NULL
2	DERIVED	l	NULL	ref	teacher_id,student_id	student_id	4	const	3	100.00	Using temporary
2	DERIVED	t	NULL	eq_ref	PRIMARY,teachers_i...	PRIMARY	4	ege_online_school.l.teacher_id	1	100.00	Using index

Рис. 19: Результат EXPLAIN шестого запроса

Объяснение запроса

Во внешнем запросе выбираются имена учителей (teacher_name) и студента (student_name). Здесь используются LEFT JOIN, чтобы включить учителей, которые не связаны с определенным студентом (student_id = 208269). Важно отметить, что студент связан с таблицей students, а учитель с таблицей teachers.

Внутри подзапроса (subquery) выбираются учителя (teacher_id), которые связаны с учебными уроками (lessons), на которых присутствует студент с идентификатором 208269. Это делается с помощью INNER JOIN между таблицами teachers и lessons, где teacher_id совпадает и student_id равно 208269. Используется DISTINCT, чтобы получить уникальных учителей.

Затем выполняется первый LEFT JOIN, который соединяет внешний запрос с таблицей teachers. Соединение выполняется по teacher_id, и это LEFT JOIN означает, что все учителя из таблицы teachers будут включены в результат, даже если они не связаны с учителем из подзапроса.

Второй LEFT JOIN связывает результат первого LEFT JOIN с таблицей students, чтобы получить имена студента.

В конце запроса используется WHERE subquery.teacher_id IS NULL для фильтрации результатов. Это означает, что будут выбраны только те строки, где teacher_id из подзапроса равен NULL, что указывает на учителей, которые не связаны со студентом 208269.

3.8 Запрос №7

Формулировка запроса

Для каждого предмета вывести первые 50 учителей и количество уроков, которые они ведут.

Код запроса

```
    SELECT t.teacher_id, t.teacher_name, s.subject_id, COUNT(l.lesson_id) AS num_lessons
FROM teachers t
CROSS JOIN subjects s
LEFT JOIN lessons l ON t.teacher_id = l.teacher_id AND s.subject_id = l.subject_id
GROUP BY t.teacher_id, t.teacher_name, s.subject_id
ORDER BY t.teacher_id, s.subject_id
LIMIT 250;
```

Результат запроса

Запрос выполнился за 6172 миллисекунд. В результирующей таблице получилось 250 строк, из-за установленного Limit = 250.

teacher_id	teacher_name	subject_id	num_lessons
1	Максим Осипов	1	1
1	Максим Осипов	2	0
1	Максим Осипов	3	0
1	Максим Осипов	4	0
1	Максим Осипов	5	0
2	Виталий Лебедев	1	0
2	Виталий Лебедев	2	0
2	Виталий Лебедев	3	0
2	Виталий Лебедев	4	0
2	Виталий Лебедев	5	0
3	Константин Пет...	1	2
3	Константин Пет...	2	0
3	Константин Пет...	3	0
3	Константин Пет...	4	0
3	Константин Пет...	5	0

Рис. 20: Результат седьмого запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	s	NULL	index	NULL	subject	4	NULL	5	100.00	Using index; Using temporary; Using filesort
1	SIMPLE	t	NULL	ALL	NULL	NULL	NULL	NULL	119570	100.00	Using join buffer (hash join)
1	SIMPLE	l	NULL	ref	teacher_id,subject_id	teacher_id	4	ege_online_school.t.teacher_id	1	100.00	Using where

Рис. 21: Результат EXPLAIN седьмого запроса

Гистограмма, построенная по запросу

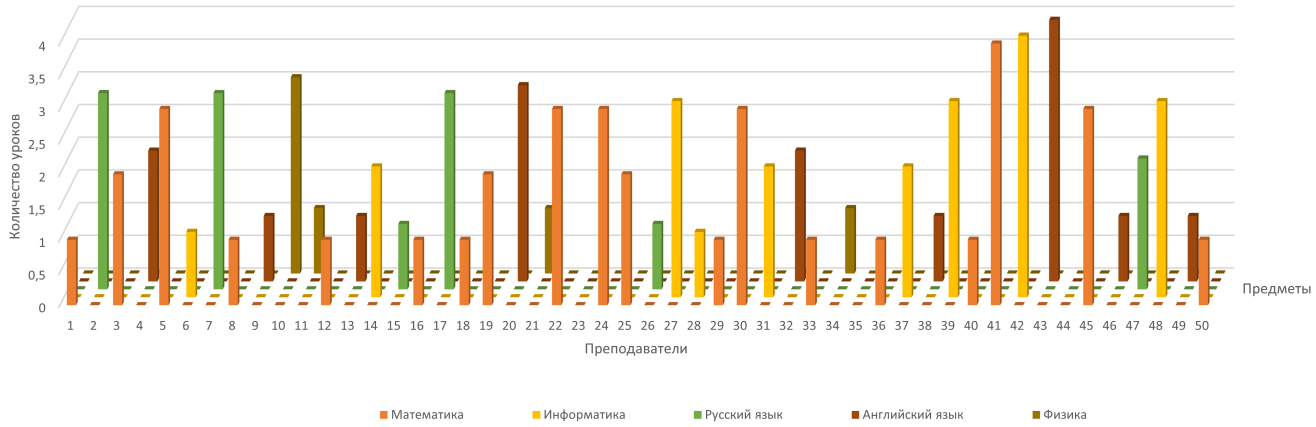

Рис. 22: Гистограмма седьмого запроса

Объяснение запроса

Запрос выполняет декартово произведение (CROSS JOIN) между таблицами teachers и subjects. Это означает, что каждая запись из таблицы teachers будет комбинироваться с каждой записью из таблицы subjects, создавая все возможные комбинации учителей и предметов. Поэтому количество строк после этого шага будет равно произведению числа учителей и числа предметов.

Затем выполняется LEFT JOIN между результатами CROSS JOIN и таблицей lessons. В этом JOIN связываются учителя (teacher_id), предметы (subject_id) и уроки (lesson_id). LEFT JOIN используется, чтобы включить все комбинации учителей и предметов, даже если нет соответствующих уроков.

Затем происходит группировка результатов по teacher_id (идентификатор учителя), teacher_name (имя учителя) и subject_id (идентификатор предмета).

Для каждой группы рассчитывается количество уроков (lesson_id), в которых участвует данный учитель и предмет. Это делается с помощью COUNT(l.lesson_id).

Результаты сортируются в порядке возрастания teacher_id и subject_id с использованием оператора ORDER BY.

Наконец, используется оператор LIMIT 250, чтобы ограничить вывод до 250 строк.

3.9 Запрос №8.1

Формулировка запроса

Для каждого предмета вывести число вебинаров.

Код запроса

```
    SELECT
  s.subject_name,
```

```
  COUNT(web.webinar_id) AS num_webinars
FROM subjects s
LEFT JOIN webinars web ON s.subject_id = web.subject_id
GROUP BY s.subject_id, s.subject_name
ORDER BY s.subject_id;
```

Результат запроса

Запрос выполнился за 0 миллисекунд. В результирующей таблице получилось 5 строк.

subject_name	num_webinars
Математика	19
Информатика	28
Русский язык	28
Английский язык	48
Физика	30

Рис. 23: Результат 8.1 запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	s	NULL	ALL	NULL	NULL	NULL	NULL	5	100.00	Using temporary; Using filesort
1	SIMPLE	web	NULL	ref	webinars_ibfk_1	webinars_ibfk_1	4	ege_online_school.s.subject_id	30	100.00	Using index

Рис. 24: Результат EXPLAIN 8.1 запроса

Объяснение запроса

Извлекаются названия предметов (subject _name) из таблицы subjects.

Производится LEFT JOIN между таблицей subjects (s) и таблицей webinars (web) по полю subject_id, чтобы связать предметы с вебинарами. LEFT JOIN используется, чтобы включить все предметы, даже если для них нет соответствующих вебинаров.

Результаты группируются с использованием оператора GROUP BY по subject_id и subject_name. Это означает, что строки будут объединены в группы на основе subject_id, и для каждой группы будет рассчитано количество вебинаров (num_webinars).

Результаты сортируются в порядке возрастания subject_id с использованием оператора ORDER BY. Это означает, что строки будут отсортированы по идентификатору предмета (subject_id).

3.10 Запрос №8.2

Формулировка запроса

Для каждого предмета вывести число уроков.

Код запроса

```
   SELECT
  s.subject_name,
  COUNT(les.lesson_id) AS num_lessons
FROM subjects s
LEFT JOIN lessons les ON s.subject_id = les.subject_id
GROUP BY s.subject_id, s.subject_name
ORDER BY s.subject_id;
```

Результат запроса

Запрос выполнился за 453 миллисекунд. В результирующей таблице получилось 5 строк.

subject_name	num_lessons
Математика	59313
Информатика	35364
Русский язык	35329
Английский язык	35146
Физика	34848

Рис. 25: Результат 8.2 запроса

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	s	NULL	ALL	NULL	NULL	NULL	NULL	5	100.00	Using temporary; Using filesort
1	SIMPLE	les	NULL	ref	subject_id	subject_id	4	ege_online_school.s.subject_id	45872	100.00	Using index

Рис. 26: Результат EXPLAIN 8.2 запроса

Гистограмма, построенная по запросу

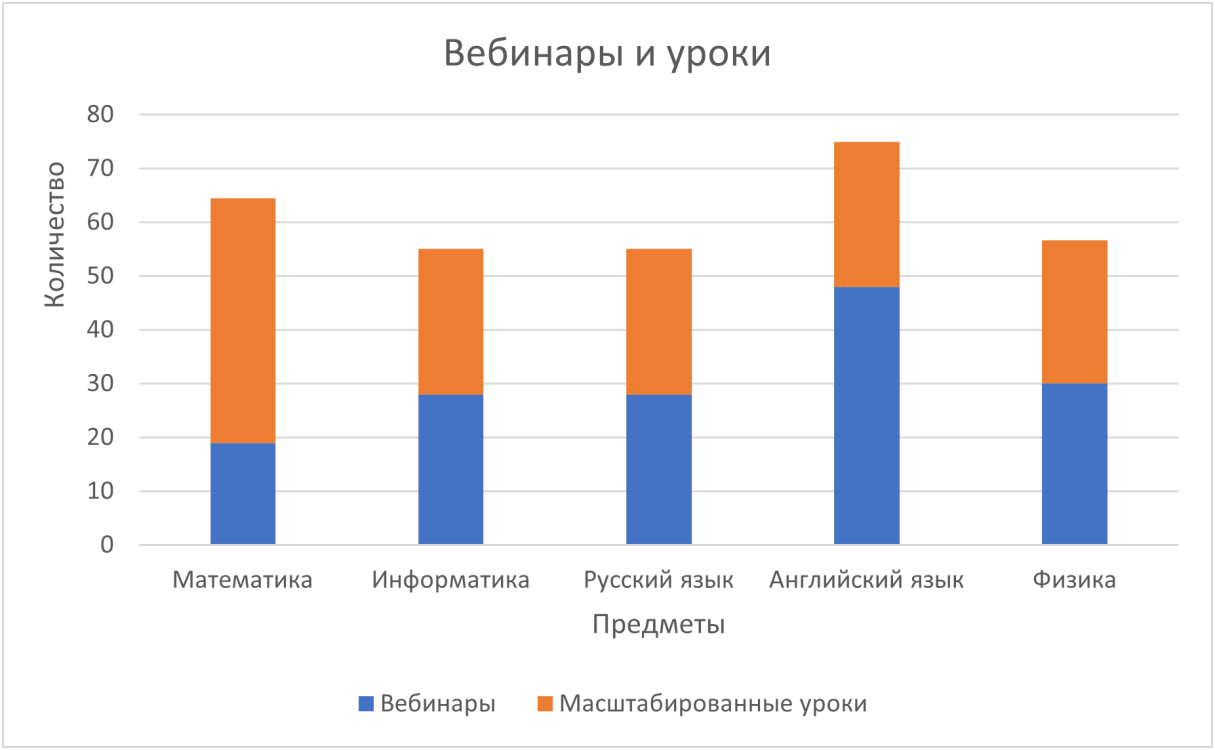

Рис. 27: Гистограмма объединяющая запросы 8.1 и 8.2

Объяснение запроса

Извлекаются названия предметов (subject_name) из таблицы subjects (s).

Производится LEFT JOIN между таблицей subjects (s) и таблицей lessons (les) по полю subject_id, чтобы связать предметы с уроками. LEFT JOIN используется, чтобы включить все предметы, даже если для них нет соответствующих уроков в таблице lessons.

Результаты группируются с использованием оператора GROUP BY по subject_id и subject_name. Это означает, что строки будут объединены в группы на основе subject_id, и для каждой группы будет рассчитано количество уроков (num_lessons).

Результаты сортируются в порядке возрастания subject_id с использованием оператора ORDER BY. Это означает, что строки будут отсортированы по идентификатору предмета (subject_id).