

Лабораторная работа № 5 Индексы, хранимые процедуры, представления, триггеры	Ф.И.О.	Боков Святослав Дмитриевич
	Группа	ИВТ-263
	Преподаватель	Аль-Мерри Гаис Мохаммед Салех
	Дата сдачи	25.04.2024

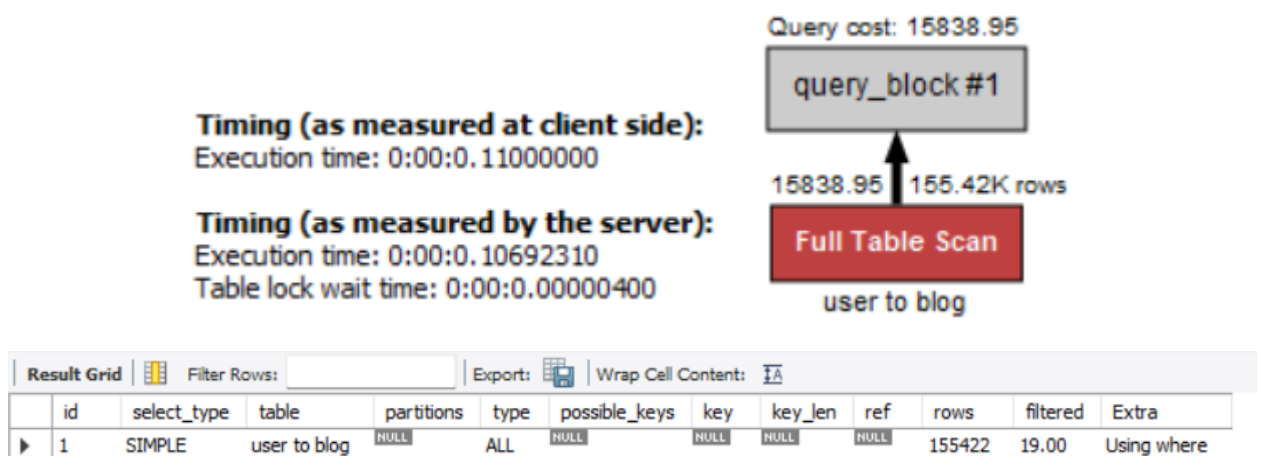
В каждую таблицу базы данных было добавлено более 10000 записей.

## Индексы

### 1. Индекс idx\_privilege

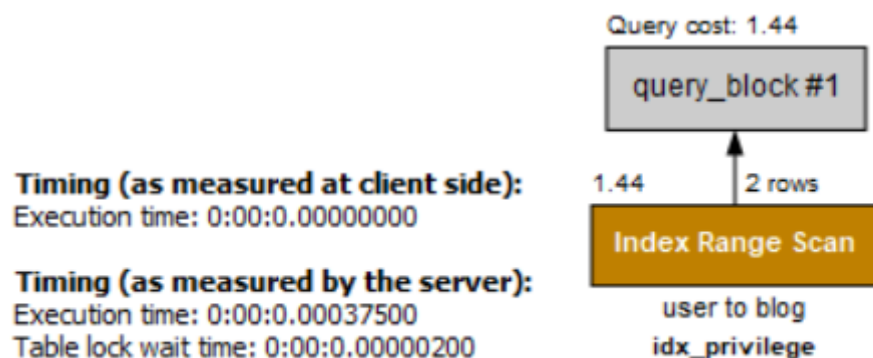
```
-- 1. Выбрать всех связи между блогами и пользователями, в которых пользователь
-- является или редактором или создателем
SELECT *
FROM `user to blog`
WHERE Privilege = 'Редактор' OR Privilege = 'Создатель';
```

До добавления индекса:



После добавления индекса на поле Privilege таблицы `user to blog`:

CREATE INDEX idx\_privilege ON `user to blog`(Privilege);

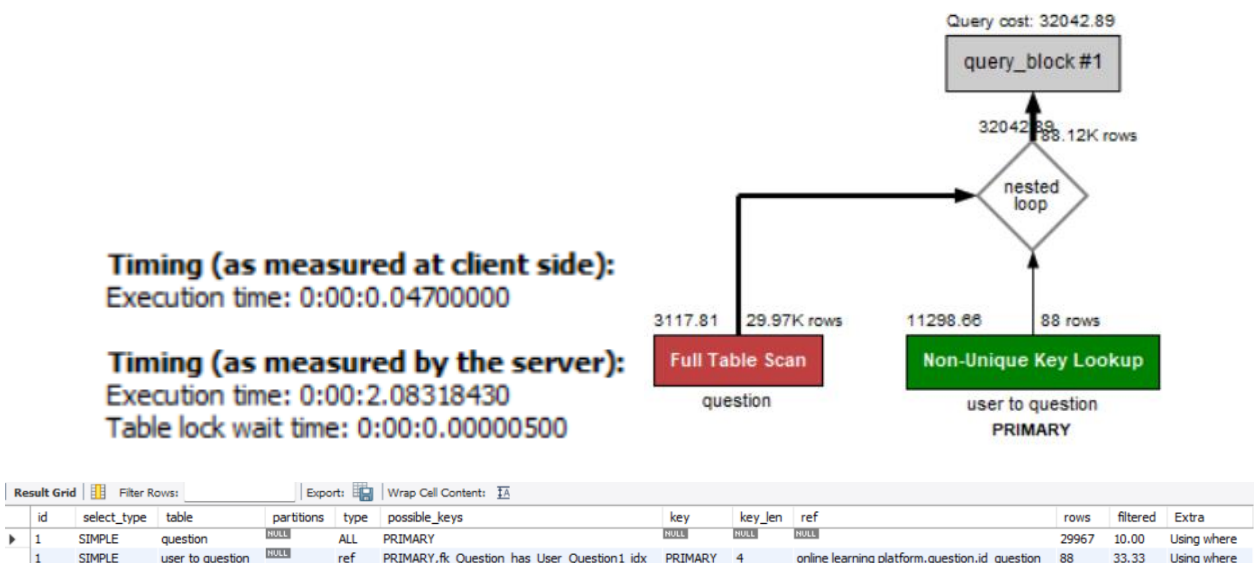


Result Grid												
	id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
▶	1	SIMPLE	user to blog	HULL	range	idx_privilege	idx_privilege	137	HULL	2	100.00	Using where; Using index

## 2. Индекс idx\_question\_type

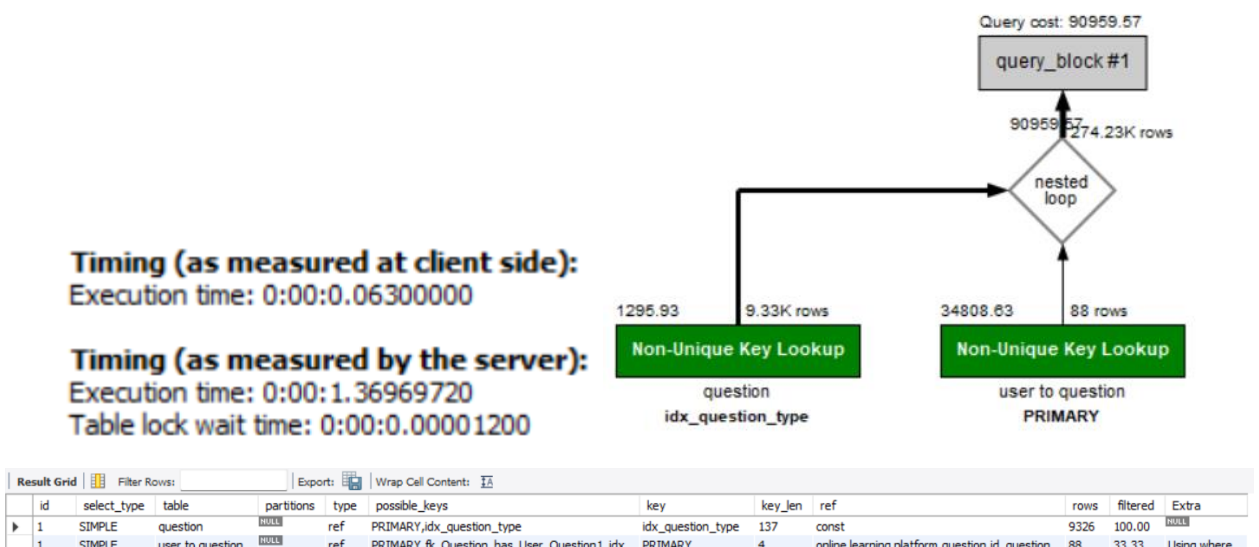
```
-- 2. Выбрать все баллы выше 5 которые получили пользователи отвечая на вопросы типа "Короткий ответ"
SELECT *
FROM `user to question`
JOIN question ON question.id_question = Question_id_question
WHERE The_score_received > 5 AND question.Type = "Короткий ответ"
LIMIT 0, 4000000;
```

До добавления индекса:



После добавления индекса на поле Type таблицы question:

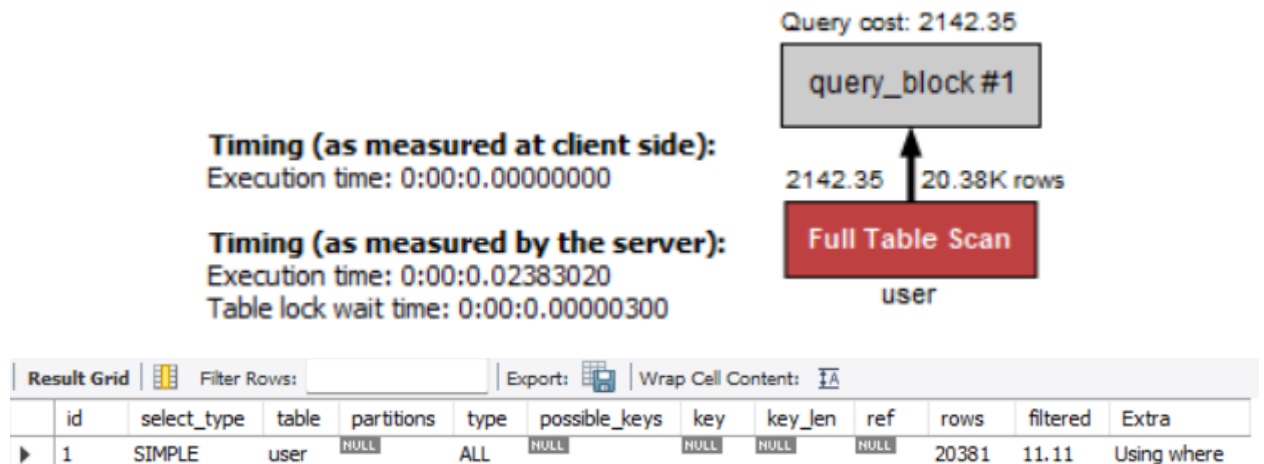
CREATE INDEX idx\_question\_type ON question(Type);



### 3. Индекс idx\_email

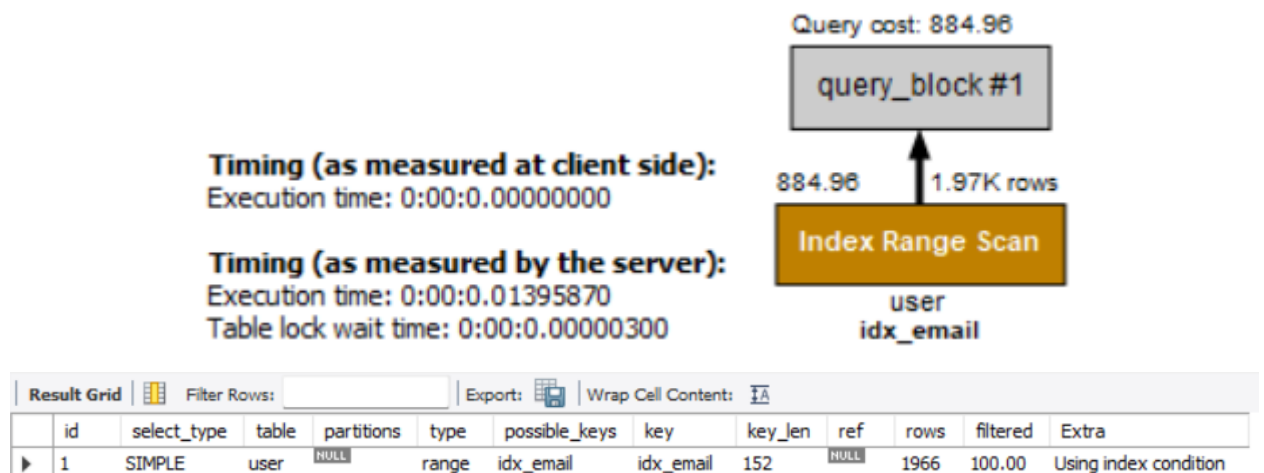
```
-- 3. Выбрать всех пользователей, у которых mail начинается на букву "a"
SELECT *
FROM user
WHERE mail LIKE "a%"
LIMIT 0,40000;
```

До добавления индекса:



После добавления индекса на поле mail таблицы user:

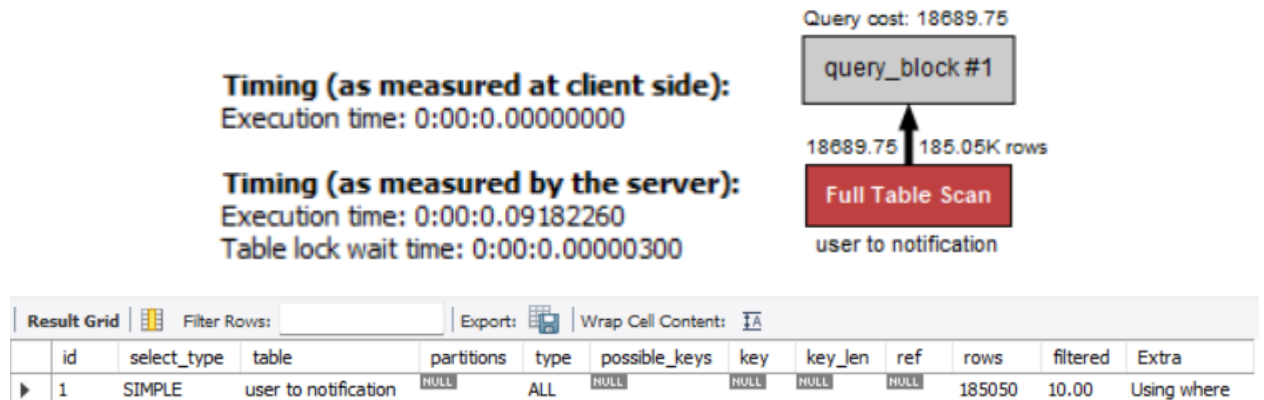
CREATE INDEX idx\_email ON user(mail);



### 4. Индекс idx\_read

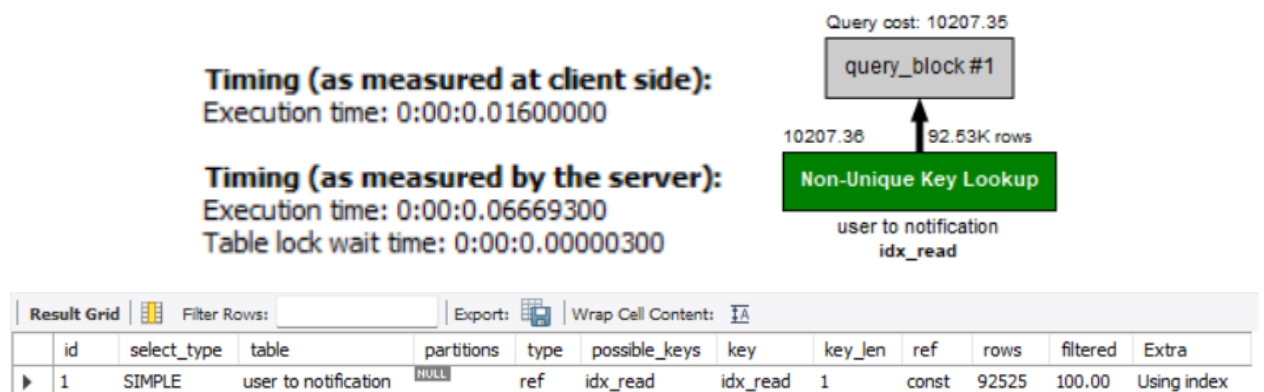
```
-- 4. Выбрать все непроиндексированные уведомления
SELECT *
FROM `user to notification`
WHERE `Read` = 0
LIMIT 0,4000000;
```

До добавления индекса:



После добавления индекса на поле Read таблицы `user to notification`:

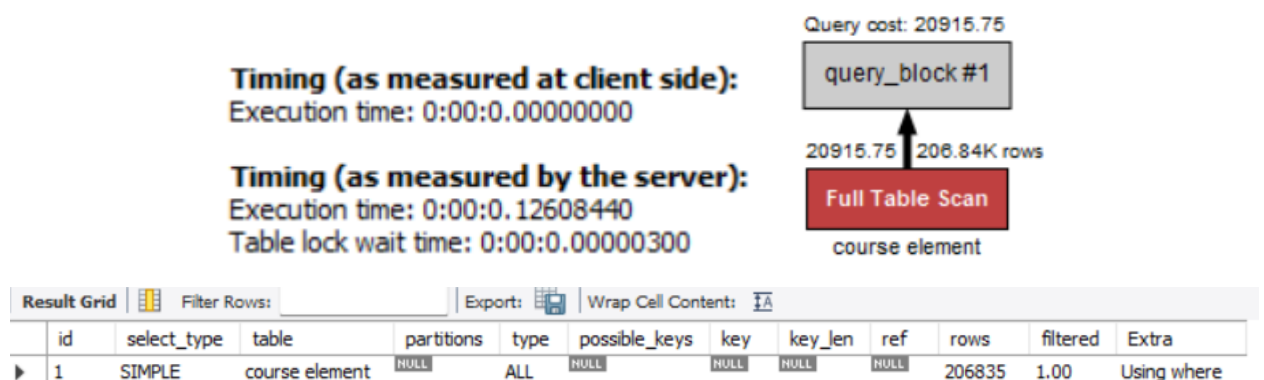
CREATE INDEX idx\_read ON `user to notification`(`Read`);



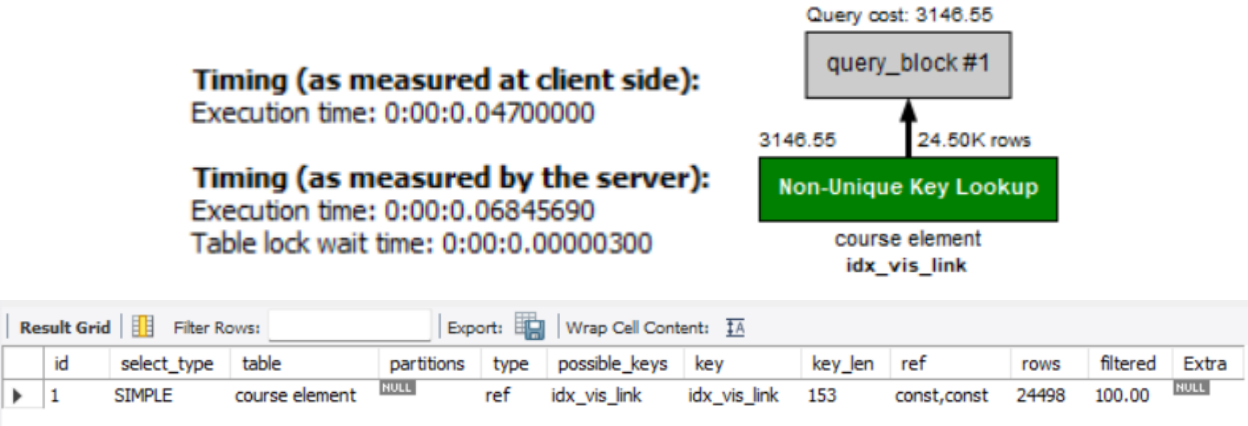
## 5. Индекс idx\_vis\_link

```
-- 5. Выбрать все скрытые элементы курсов типа "Ссылка"
SELECT *
FROM `course element`
WHERE Visibility = 0
      AND Type = "Ссылка"
LIMIT 0,4000000;
```

До добавления индекса:



После добавления индекса на поля Visibility и Type таблицы `course element`:  
CREATE INDEX idx\_vis\_link ON `course element`(Visibility, Type);



## Функции и процедуры

### 1. Функция get\_user\_age

```
-- 1. Рассчитать возраст пользователя
DELIMITER $$
CREATE FUNCTION get_user_age(date_of_birth date)
RETURNS INT
BEGIN
    DECLARE age_user INT;
    SET age_user = TIMESTAMPDIFF(YEAR, date_of_birth, CURRENT_DATE);
    RETURN age_user;
END $$
DELIMITER ;
```

Запрос с использованием функции:

```
SELECT Login,
       Date_of_birth,
       Mail,
       get_user_age(Date_of_birth) as age
FROM user;
```

Результат запроса:

	Login	Date_of_birth	Mail	age
►	Aaliyah.Baumbach56	2009-06-09	Chaim50@gmail.com	14
	Aaliyah.Durgan48	1983-12-28	Jaren.Dibbert@hotmail.com	40
	Aaliyah.Olson26	2000-12-10	Oma90@yahoo.com	23
	Aaliyah41	2000-10-24	Elna_Balistreri@gmail.com	23
	Aaron.Ernser	1983-03-04	Carson47@gmail.com	41
	Aaron72	1989-03-26	Haylie85@gmail.com	35
	Aaron_Hyatt	1999-10-22	Lonzo88@hotmail.com	24
	Aaron_Kirlin	2008-07-05	Stefanie_Bogan92@yahoo.com	15
	Aaron_Prohaska30	2013-02-26	Domenica0@gmail.com	11
	Abigail.Borer	2011-04-19	Alessia.Marvin@hotmail.com	13
	Abigail.Conn51	2012-03-10	Stacey_Green@yahoo.com	12
	Abigail.Reynolds	2004-07-23	Kristoffer.Cummings@hotmail....	19
	Abigail44	1991-05-16	Elenora_Sawayn@gmail.com	32
	Abigail95	1999-06-12	Judy_Quitzon2@hotmail.com	24
	Abigail_DuRue96	1986-08-07	Trace8@hotmail.com	37

## 2. Функция get\_rang

```
-- 2. Вывод рейтинга курса (если рейтинг >= 0.475, то вывести "высокий",
-- если рейтинг >= 0.45, то вывести "средний", иначе - "низкий")
DELIMITER $$
CREATE FUNCTION get_rang(rating FLOAT)
RETURNS VARCHAR(50)
BEGIN
    DECLARE rang VARCHAR(50);
    SET rang = IF(rating >= 0.475, "высокий", IF(rating >= 0.45, "средний", "низкий"));
    RETURN rang;
END $$
DELIMITER ;
```

Запрос с использованием функции:

```
SELECT `Short name`,
       Rating,
       Duration,
       get_rang(Rating) as StringRating
FROM course;
```

Результат запроса:

	Short name	Rating	Duration	StringRating
	ai-1-s	0.31	3477	низкий
	ai-1-z	0.31	21425	низкий
	ai-10-g	0.37	29901	низкий
	ai-100-s	0.48	28966	высокий
	ai-101-f	0.46	39793	средний
	ai-101-g	0.48	4041	высокий
	ai-101-o	0.5	12574	высокий
	ai-101-u	0.42	14026	низкий
	ai-101-x	0.3	30997	низкий

## 3. Функция get\_visibility

```
-- 3. Вывод доступности через видимость элемента
DELIMITER $$
CREATE FUNCTION get_visibility(vis int)
RETURNS VARCHAR(50)
BEGIN
    DECLARE visible VARCHAR(50);
    set visible = CASE vis
        WHEN 0 THEN "Скрытый"
        WHEN 1 THEN "Открытый"
    END;
    RETURN visible;
END $$
DELIMITER ;
```

Запрос с использованием функции:



```

SELECT id_course_element,
       Number,
       Type,
       Visibility,
       get_visibility(Visibility) as StringVisibility
FROM `course_element`;

```

Результат запроса:

	id_course_element	Number	Type	Visibility	StringVisibility
▶	1	1	Файл	0	Скрытый
	2	2	Видеовстреча	0	Скрытый
	3	3	Исходный код	1	Открытый
	4	4	Документ	1	Открытый
	5	5	Видеовстреча	0	Скрытый
	6	6	Документ	1	Открытый
	7	7	Диаграмма	1	Открытый

#### 4. Процедура GetMaxPrice

```

-- 4. Вывести цену самого дорогого курса на платформе
DELIMITER $$
CREATE PROCEDURE GetMaxPrice()
BEGIN
    SELECT course.Name,
           payment.Price
    FROM course
    JOIN payment ON payment.id_payment = course.Payment_id_payment
    WHERE payment.Price = (SELECT MAX(Price) FROM payment);
END $$
DELIMITER ;

```

Вызов процедуры:

```
CALL GetMaxPrice();
```

Результат вызова:

	Name	Price
▶	front course at 122	9999
	vision course at 21	9999
	front course at 133	9999
	xyz course at 167	9999
	qwerty course at 188	9999
	cpp course at 95	9999



## 5. Процедура GetMail

```
-- 5. Выбор всех имён пользователей с почтой, у которых почта удовлетворяет шаблону
DELIMITER $$
CREATE PROCEDURE GetMail(shablon VARCHAR(50))
BEGIN
    SELECT Name,
           mail
    FROM user
    WHERE mail LIKE shablon;
END $$
DELIMITER ;
```

Вызов процедуры:

```
CALL GetMail("a%gmail.com");
```

Результат вызова:

	Name	mail
►	Wilma Bruen	Alexandre55@gmail.com
	Susie Hickie Sr.	Angelo.Stroman38@gmail.com
	Eduardo Nienow	Alysa98@gmail.com
	Ron Luetttgen IV	Audreanne_Feeney@gmail.com
	Ms. Nadine Ortiz	Anika_Weimann@gmail.com
	Rodolfo O'Kon	Alene.Lesch@gmail.com

## 6. Процедура GetTestingEndInfo

```
-- 6. Вывод часа, дня, недели окончания каждого тестирования
DELIMITER $$
CREATE PROCEDURE GetTestingEndInfo()
BEGIN
    SELECT id_test, HOUR(End_date) AS hour, DAY(End_date) AS day, WEEK(End_date) AS week
    FROM testing;
END $$
DELIMITER ;
```

Вызов процедуры:

```
CALL GetTestingEndInfo();
```

Результат вызова:

	id_test	hour	day	week
►	1	20	21	42
	2	0	14	33
	3	6	17	29
	4	5	7	19
	5	10	13	41
	6	22	20	20
	7	22	9	32

## 7. Процедура course\_price

-- 7. Вывод суммарной стоимости какого-то количества курсов, отсортированных по рейтингу

DELIMITER \$\$

CREATE PROCEDURE course\_price(lim INT)

BEGIN

DECLARE summa, summ, counter INT;

set summa = 0;

set counter = 0;

label: LOOP

IF counter = lim THEN

LEAVE label;

END IF;

SELECT payment.price FROM course

JOIN payment ON payment.id\_payment = course.Payment\_id\_payment

ORDER BY Rating DESC

LIMIT 1 OFFSET counter INTO summ;

SET summa = summa + summ;

SET counter = counter + 1;

END LOOP label;

SELECT summa;

END \$\$

DELIMITER ;

Вызов процедуры:

CALL course\_price(6);

Результат вызова:

	summa
▶	23495

## 8. Процедура

```
-- 8. Вывести все элементы указанного курса в порядке их следования в курсе (вместо
-- NULL в прикрепленных файлах выводить "N/A")
DELIMITER $$
CREATE PROCEDURE course_elements(course_short_name VARCHAR(50))
BEGIN
    DECLARE id_content_bank INT;

    SELECT `Content bank_id_content_bank`
    FROM course
    WHERE course_short_name = `Short name`
    INTO id_content_bank;

    SELECT Number, Name, Type, Visibility, IFNULL(Attached_file, 'N/A') AS File
    FROM `course element`
    WHERE `Content bank_id_content_bank` = id_content_bank
    ORDER BY Number;

END $$
DELIMITER ;
```

Вызов процедуры:

```
CALL course_elements("ai-1-s");
```

Результат вызова:

	Number	Name	Type	Visibility	File
►	1	uuioyolxqxiwi	Исходный код	0	N/A
	2	vaibeafibezbn	Видеовстреча	0	N/A
	3	ababbhitdrkjstyl	Диаграмма	0	N/A
	4	ithaqdcmrpsbcnl	Исходный код	1	bank_content/course_element/4
	5	nappasbjmuxabkfe	Ссылка	1	N/A
	6	pjfxrtawgzlscpi	PDF	0	N/A
	7	xdkmguiupymremjujaq	Исходный код	0	N/A
	8	jpglzbwbmkvravqbnb	Видеовстреча	1	bank_content/course_element/8

## Представления

### 1. Представление course\_info

```
-- 1. Представление для полной информации о курсах
CREATE VIEW course_info
AS SELECT Name AS course_name,
        Duration,
        The_beginning_of_the_course,
        Rating,
        forum.Number_of_messages,
        payment.Price,
        payment.Commission,
        `content bank`.Location_on_the_server
FROM course
JOIN forum ON forum.id_forum = course.Forum_id_forum
JOIN payment ON payment.id_payment = Payment_id_payment
JOIN `content bank` ON `content bank`.id_content_bank = `Content bank_id_content_bank`
ORDER BY The_beginning_of_the_course, course_name;
```

Запрос:

```
SELECT *
FROM course_info
WHERE Price > 7000;
```

Результат запроса:

	course_name	Duration	The_beginning_of_the_course	Rating	Number_of_messages	Price	Commission	Location_on_the_server
►	ai course at 35	45308	2021-01-01 14:14:24	0.5	51	8999	0.01	/courses/bank1550
	xyz course at 73	9756	2021-01-01 17:14:00	0.33	55	7999	0.16	/courses/bank2329
	analyse course at 58	4308	2021-01-01 20:40:41	0.3	62	8999	0.09	/courses/bank11912
	n course at 178	13089	2021-01-02 00:00:45	0.37	84	8999	0.15	/courses/bank8014
	vision course at 91	27961	2021-01-02 00:27:55	0.31	82	9499	0.12	/courses/bank10971
	comp course at 120	10251	2021-01-02 09:32:58	0.34	3	9499	0.11	/courses/bank7034
	back course at 85	13401	2021-01-02 10:30:28	0.35	38	7999	0.07	/courses/bank10807
	ai course at 173	5463	2021-01-02 10:50:24	0.33	93	7999	0.18	/courses/bank10342
	analyse course at 154	28064	2021-01-02 22:31:38	0.44	42	8999	0.08	/courses/bank279

### 2. Представление user\_score

```
-- 2. Получить список пользователей и их оценок, включая тех, у которых нет оценок
CREATE VIEW user_score
AS SELECT user.Login,
        user.Country,
        `user to question`.The_score_received,
        question.id_question,
        question.Score_for_correct_execution,
        question.Type
FROM question
RIGHT JOIN `user to question` ON `user to question`.Question_id_question = question.id_question
RIGHT JOIN user ON user.Login = `user to question`.User_Login;
```

Запрос:

```
SELECT * FROM user_score
WHERE The_score_received * 2 > Score_for_correct_execution;
```

Результат запроса:

	Login	Country	The_score_received	id_question	Score_for_correct_execution	Type
►	Carmine.Schumm53	Armenia	10.08	1	14.65	Сопоставьте варианты
	Erik.Ward	Micronesia	10.51	1	14.65	Сопоставьте варианты
	Gianni_Cronin	Maldives	11.95	1	14.65	Сопоставьте варианты
	Henry40	American Samoa	11.36	1	14.65	Сопоставьте варианты
	Ignacio11	Antigua and Barbuda	12.71	1	14.65	Сопоставьте варианты
	Jeff.Green	Saint Lucia	11.04	1	14.65	Сопоставьте варианты
	Kevin.McGlynn77	Russian Federation	11.97	1	14.65	Сопоставьте варианты
	Laila_Lynch76	Vietnam	8.1	1	14.65	Сопоставьте варианты
	Liza_Bogan	Canada	9.02	1	14.65	Сопоставьте варианты

### 3. Представление test\_question

```
-- 3. Вывести список вопросов для каждого теста (в котором даже нет вопросов), отсортированных по номеру вопроса в тесте
CREATE VIEW test_question
AS SELECT id_test,
        `testing to question`.Question_id_question AS id_question,
        `testing to question`.Number AS number
FROM testing
LEFT JOIN `testing to question` ON `testing to question`.Testing_id_test = id_test
ORDER BY id_test, number;
```

Запрос:

```
SELECT *
FROM test_question
WHERE id_test IN (4, 67, 135, 907, 16);
```

Результат запроса:

	id_test	id_question	number
►	4	9760	1
	4	414	2
	4	12150	3
	16	1075	1
	16	19208	2
	16	28951	3
	16	11037	4

## Триггер

```
-- Обновление баллов студентов за вопрос при смене балла за правильное выполнение вопроса
-- (доля полученного балла должна оставаться прежней)
DELIMITER $$
CREATE TRIGGER user_score AFTER UPDATE ON question
FOR EACH ROW
BEGIN
    UPDATE `user to question`
    SET The_score_received = (The_score_received /
    (SELECT OLD.Score_for_correct_execution
    FROM question WHERE id_question = OLD.id_question) * 100) div 1 / 100 * NEW.Score_for_correct_execution
    WHERE Question_id_question = OLD.id_question;
END $$
DELIMITER ;
```

Тест работы триггера (сначала балл за изменяемый вопрос был равен 50, затем изменился на 30, потом опять изменяем на 50):

```
SELECT *
FROM `user to question`
WHERE Question_id_question = 5
LIMIT 7;
```

```
SELECT *
FROM question
WHERE id_question = 5;
```

```
UPDATE question
SET Score_for_correct_execution = 30
WHERE id_question = 5;
```

```
SELECT *
FROM `user to question`
WHERE Question_id_question = 5
LIMIT 7;
```

```
SELECT *
FROM question
WHERE id_question = 5;
```

```
UPDATE question
SET Score_for_correct_execution = 50
WHERE id_question = 5;
```

Результат тестирования:

	Question_id_question	User_Login	The_score_received
▶	5	Aaliyah.Olson26	38.5
	5	Adaline_Treutel98	21
	5	Adelbert.Abernathy67	21
	5	Adell_Bernier21	2.5
	5	Agustin.Barton84	45
	5	Agustin28	16
	5	Agustina_Wiza66	41
▲	NULL	NULL	NULL

	id_question	Name	Text	Score_for_correct_execution	Creation_time	Type	The_maximum_size	The_maxin
▶	5	fzxgfyb	Ilo expedita magnam...	50	2023-12-01 18:32:29	Программа	59	15
▲	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

	Question_id_question	User_Login	The_score_received
▶	5	Aaliyah.Olson26	23.1
	5	Adaline_Treutel98	12.6
	5	Adelbert.Abernathy67	12.6
	5	Adell_Bernier21	1.5
	5	Agustin.Barton84	27
	5	Agustin28	9.6
	5	Agustina_Wiza66	24.6
▲	NULL	NULL	NULL

	id_question	Name	Text	Score_for_correct_execution	Creation_time	Type	The_maximum_size	The_max
▶	5	fzxgfyb	Ilo expedita magnam...	30	2023-12-01 18:32:29	Программа	59	15
▲	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL