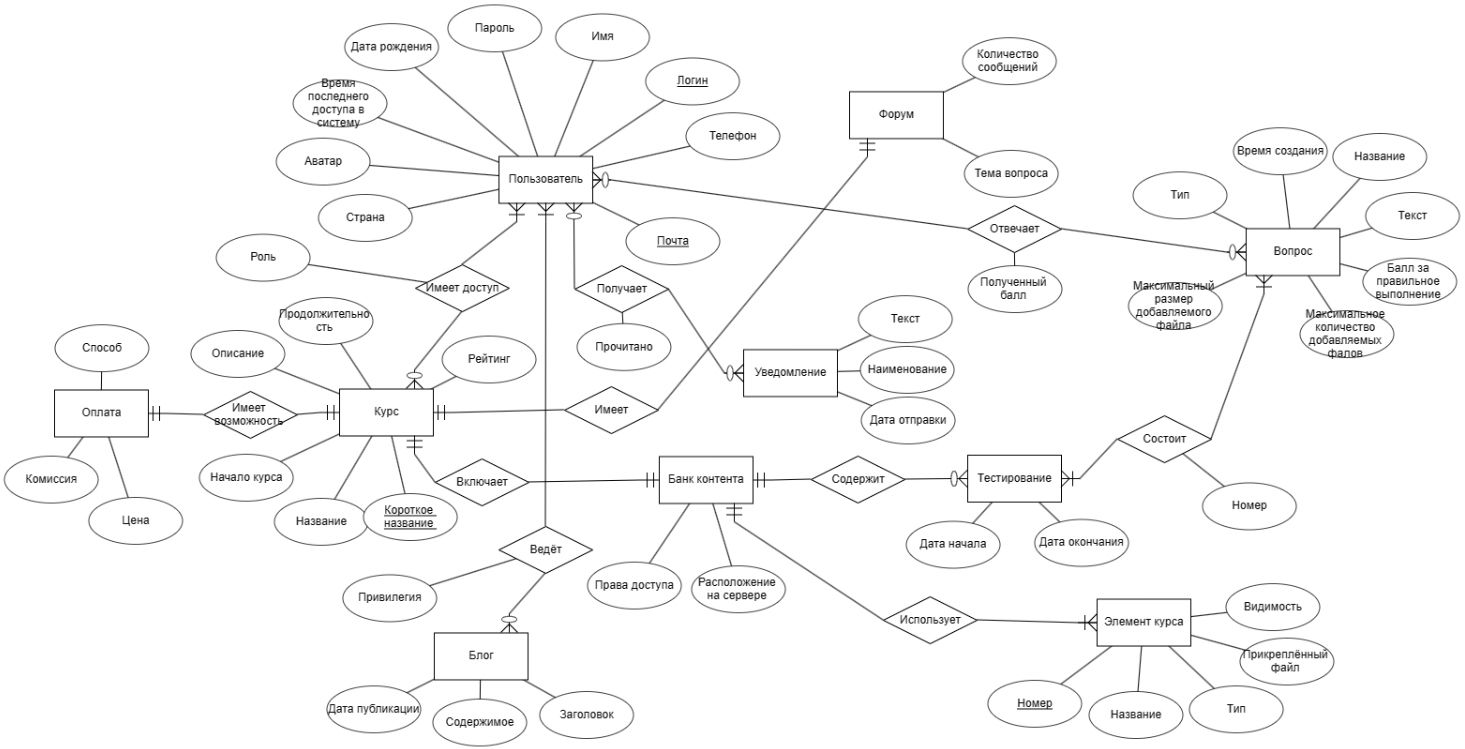
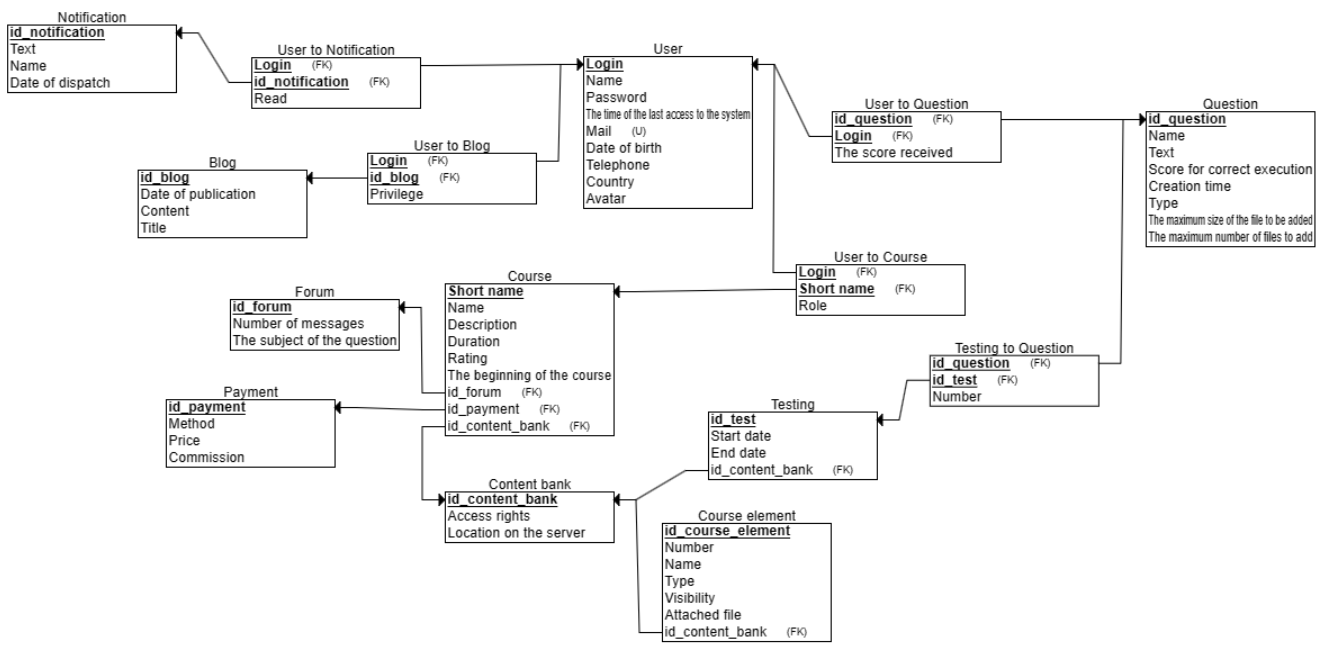
|  |  |  |
| --- | --- | --- |
| Лабораторная работа № 3  Физическое проектирование базы данных | Ф.И.О. | Боков Святослав Дмитриевич |
| Группа | ИВТ-263 |
| Преподаватель | Аль-Мерри Гаис Мохаммед Салех |
| Дата сдачи | 14.03.2024 |

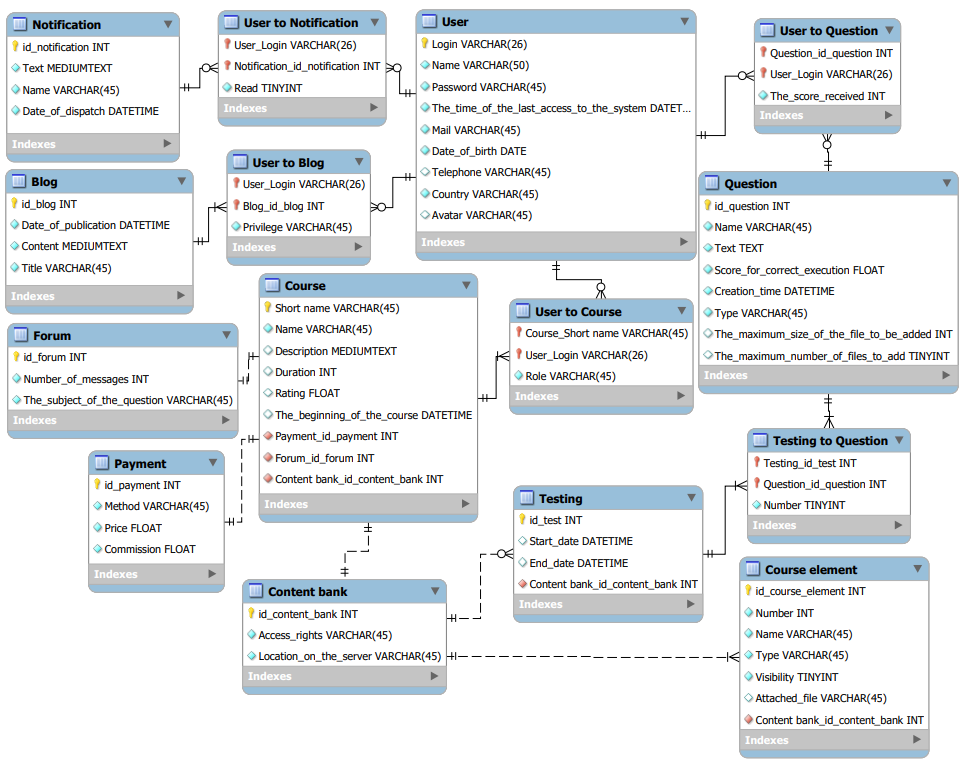
**ER-диаграмма из первой лабораторной работы:**



**Реляционная схема из второй лабораторной:**



**Реляционная схема в MySQL**:



**SQL скрипт для создания базы данных и таблиц:**

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema Online learning platform

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `Online learning platform` DEFAULT CHARACTER SET utf8 ;

USE `Online learning platform` ;

-- -----------------------------------------------------

-- Table `Online learning platform`.`User`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`User` (

`Login` VARCHAR(26) NOT NULL,

`Name` VARCHAR(50) NOT NULL,

`Password` VARCHAR(45) NOT NULL,

`The\_time\_of\_the\_last\_access\_to\_the\_system` DATETIME NOT NULL,

`Mail` VARCHAR(45) NOT NULL,

`Date\_of\_birth` DATE NOT NULL,

`Telephone` VARCHAR(45) NULL,

`Country` VARCHAR(45) NOT NULL,

`Avatar` VARCHAR(45) NULL,

PRIMARY KEY (`Login`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Notification`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Notification` (

`id\_notification` INT NOT NULL AUTO\_INCREMENT,

`Text` MEDIUMTEXT NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Date\_of\_dispatch` DATETIME NOT NULL,

PRIMARY KEY (`id\_notification`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Blog`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Blog` (

`id\_blog` INT NOT NULL AUTO\_INCREMENT,

`Date\_of\_publication` DATETIME NOT NULL,

`Content` MEDIUMTEXT NOT NULL,

`Title` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id\_blog`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Payment`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Payment` (

`id\_payment` INT NOT NULL AUTO\_INCREMENT,

`Method` VARCHAR(45) NOT NULL,

`Price` FLOAT NOT NULL,

`Commission` FLOAT NOT NULL,

PRIMARY KEY (`id\_payment`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Forum`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Forum` (

`id\_forum` INT NOT NULL AUTO\_INCREMENT,

`Number\_of\_messages` INT NOT NULL,

`The\_subject\_of\_the\_question` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id\_forum`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Content bank`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Content bank` (

`id\_content\_bank` INT NOT NULL AUTO\_INCREMENT,

`Access\_rights` VARCHAR(45) NOT NULL,

`Location\_on\_the\_server` VARCHAR(45) NOT NULL,

PRIMARY KEY (`id\_content\_bank`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Course`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Course` (

`Short name` VARCHAR(45) NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Description` MEDIUMTEXT NULL,

`Duration` INT NULL,

`Rating` FLOAT NULL,

`The\_beginning\_of\_the\_course` DATETIME NULL,

`Payment\_id\_payment` INT NOT NULL,

`Forum\_id\_forum` INT NOT NULL,

`Content bank\_id\_content\_bank` INT NOT NULL,

PRIMARY KEY (`Short name`),

INDEX `fk\_Course\_Payment\_idx` (`Payment\_id\_payment` ASC) VISIBLE,

INDEX `fk\_Course\_Forum1\_idx` (`Forum\_id\_forum` ASC) VISIBLE,

INDEX `fk\_Course\_Content bank1\_idx` (`Content bank\_id\_content\_bank` ASC) VISIBLE,

CONSTRAINT `fk\_Course\_Payment`

FOREIGN KEY (`Payment\_id\_payment`)

REFERENCES `Online learning platform`.`Payment` (`id\_payment`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Course\_Forum1`

FOREIGN KEY (`Forum\_id\_forum`)

REFERENCES `Online learning platform`.`Forum` (`id\_forum`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Course\_Content bank1`

FOREIGN KEY (`Content bank\_id\_content\_bank`)

REFERENCES `Online learning platform`.`Content bank` (`id\_content\_bank`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Course element`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Course element` (

`id\_course\_element` INT NOT NULL AUTO\_INCREMENT,

`Number` INT NOT NULL,

`Name` VARCHAR(45) NOT NULL,

`Type` VARCHAR(45) NOT NULL,

`Visibility` TINYINT NOT NULL,

`Attached\_file` VARCHAR(45) NULL,

`Content bank\_id\_content\_bank` INT NOT NULL,

PRIMARY KEY (`id\_course\_element`),

INDEX `fk\_Course element\_Content bank1\_idx` (`Content bank\_id\_content\_bank` ASC) VISIBLE,

CONSTRAINT `fk\_Course element\_Content bank1`

FOREIGN KEY (`Content bank\_id\_content\_bank`)

REFERENCES `Online learning platform`.`Content bank` (`id\_content\_bank`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Testing`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Testing` (

`id\_test` INT NOT NULL AUTO\_INCREMENT,

`Start\_date` DATETIME NULL,

`End\_date` DATETIME NULL,

`Content bank\_id\_content\_bank` INT NOT NULL,

PRIMARY KEY (`id\_test`),

INDEX `fk\_Testing\_Content bank1\_idx` (`Content bank\_id\_content\_bank` ASC) VISIBLE,

CONSTRAINT `fk\_Testing\_Content bank1`

FOREIGN KEY (`Content bank\_id\_content\_bank`)

REFERENCES `Online learning platform`.`Content bank` (`id\_content\_bank`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Question`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Question` (

`id\_question` INT NOT NULL AUTO\_INCREMENT,

`Name` VARCHAR(45) NOT NULL,

`Text` TEXT NOT NULL,

`Score\_for\_correct\_execution` FLOAT NOT NULL,

`Creation\_time` DATETIME NOT NULL,

`Type` VARCHAR(45) NOT NULL,

`The\_maximum\_size\_of\_the\_file\_to\_be\_added` INT NULL,

`The\_maximum\_number\_of\_files\_to\_add` TINYINT NULL,

PRIMARY KEY (`id\_question`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`User to Course`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`User to Course` (

`Course\_Short name` VARCHAR(45) NOT NULL,

`User\_Login` VARCHAR(26) NOT NULL,

`Role` VARCHAR(45) NOT NULL,

PRIMARY KEY (`Course\_Short name`, `User\_Login`),

INDEX `fk\_Course\_has\_User\_User1\_idx` (`User\_Login` ASC) VISIBLE,

INDEX `fk\_Course\_has\_User\_Course1\_idx` (`Course\_Short name` ASC) VISIBLE,

CONSTRAINT `fk\_Course\_has\_User\_Course1`

FOREIGN KEY (`Course\_Short name`)

REFERENCES `Online learning platform`.`Course` (`Short name`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Course\_has\_User\_User1`

FOREIGN KEY (`User\_Login`)

REFERENCES `Online learning platform`.`User` (`Login`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`User to Notification`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`User to Notification` (

`User\_Login` VARCHAR(26) NOT NULL,

`Notification\_id\_notification` INT NOT NULL,

`Read` TINYINT NOT NULL,

PRIMARY KEY (`User\_Login`, `Notification\_id\_notification`),

INDEX `fk\_User\_has\_Notification\_Notification1\_idx` (`Notification\_id\_notification` ASC) VISIBLE,

INDEX `fk\_User\_has\_Notification\_User1\_idx` (`User\_Login` ASC) VISIBLE,

CONSTRAINT `fk\_User\_has\_Notification\_User1`

FOREIGN KEY (`User\_Login`)

REFERENCES `Online learning platform`.`User` (`Login`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_User\_has\_Notification\_Notification1`

FOREIGN KEY (`Notification\_id\_notification`)

REFERENCES `Online learning platform`.`Notification` (`id\_notification`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`User to Blog`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`User to Blog` (

`User\_Login` VARCHAR(26) NOT NULL,

`Blog\_id\_blog` INT NOT NULL,

`Privilege` VARCHAR(45) NOT NULL,

PRIMARY KEY (`User\_Login`, `Blog\_id\_blog`),

INDEX `fk\_User\_has\_Blog\_Blog1\_idx` (`Blog\_id\_blog` ASC) VISIBLE,

INDEX `fk\_User\_has\_Blog\_User1\_idx` (`User\_Login` ASC) VISIBLE,

CONSTRAINT `fk\_User\_has\_Blog\_User1`

FOREIGN KEY (`User\_Login`)

REFERENCES `Online learning platform`.`User` (`Login`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_User\_has\_Blog\_Blog1`

FOREIGN KEY (`Blog\_id\_blog`)

REFERENCES `Online learning platform`.`Blog` (`id\_blog`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`User to Question`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`User to Question` (

`Question\_id\_question` INT NOT NULL,

`User\_Login` VARCHAR(26) NOT NULL,

`The\_score\_received` INT NOT NULL,

PRIMARY KEY (`Question\_id\_question`, `User\_Login`),

INDEX `fk\_Question\_has\_User\_User1\_idx` (`User\_Login` ASC) VISIBLE,

INDEX `fk\_Question\_has\_User\_Question1\_idx` (`Question\_id\_question` ASC) VISIBLE,

CONSTRAINT `fk\_Question\_has\_User\_Question1`

FOREIGN KEY (`Question\_id\_question`)

REFERENCES `Online learning platform`.`Question` (`id\_question`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Question\_has\_User\_User1`

FOREIGN KEY (`User\_Login`)

REFERENCES `Online learning platform`.`User` (`Login`)

ON DELETE CASCADE

ON UPDATE CASCADE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `Online learning platform`.`Testing to Question`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `Online learning platform`.`Testing to Question` (

`Testing\_id\_test` INT NOT NULL,

`Question\_id\_question` INT NOT NULL,

`Number` TINYINT NOT NULL,

PRIMARY KEY (`Testing\_id\_test`, `Question\_id\_question`),

INDEX `fk\_Testing\_has\_Question\_Question1\_idx` (`Question\_id\_question` ASC) VISIBLE,

INDEX `fk\_Testing\_has\_Question\_Testing1\_idx` (`Testing\_id\_test` ASC) VISIBLE,

CONSTRAINT `fk\_Testing\_has\_Question\_Testing1`

FOREIGN KEY (`Testing\_id\_test`)

REFERENCES `Online learning platform`.`Testing` (`id\_test`)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT `fk\_Testing\_has\_Question\_Question1`

FOREIGN KEY (`Question\_id\_question`)

REFERENCES `Online learning platform`.`Question` (`id\_question`)

ON DELETE CASCADE

ON UPDATE CASCADE)

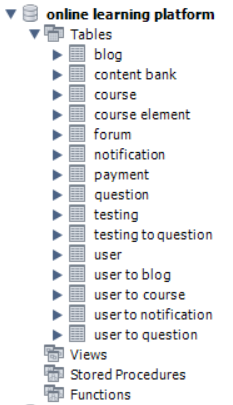
ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

**После выполнения данного SQL скрипта, получим:**



**SQL скрипт для заполнения таблиц данными:**

-- Forum

INSERT forum(id\_forum, Number\_of\_messages, The\_subject\_of\_the\_question)

VALUES

(1, 14, 'Как сделать коммит?'),

(2, 28, "Как связать удаленный репозиторий?"),

(3, 24, "Как сделать выбор данных по критерию?"),

(4, 12, "Как объединить данные из разных таблиц?"),

(5, 7, "Как добавить новую лекцию в курс?"),

(6, 33, "Как настроить автоматическую проверку кода?"),

(7, 15, "Как настроить доступ к видеолекциям?"),

(8, 16, "Как создать сертификат?"),

(9, 16, "Как отслеживать прогресс студентов?"),

(10, 20, "Как добавить новую главу в учебник?");

-- User

INSERT user(Login, Name, Password, Telephone, Mail, Country, Avatar, The\_time\_of\_the\_last\_access\_to\_the\_system, Date\_of\_birth)

VALUES

("3pknai", "Боков С.Д.", "Пупупу", "79665115211", "slavabokov2004@mail.ru", "Russia", "/files/3pknai.img", '2024-02-27 08:15', '1998-02-15'),

('mary234', 'Johnson, Mary', 'pass123', '555-1234', 'mary234@email.com', 'USA', '/files/mary234.img', '2024-02-28 09:30', '1990-06-12'),

('jdoe78', 'Doe, John', 'password1', '555-5678', 'jdoe78@gmail.com', 'Canada', '/files/jdoe78.img', '2024-02-28 15:45', '1978-03-08'),

('alex87', 'Smith, Alex', 'mysecret123', '555-9876', 'alex87@hotmail.com', 'UK', '/files/alex87.img', '2024-02-27 10:20', '1987-09-20'),

('saraB84', 'Brown, Sara', 'ilovecats12', '555-8888', 'saraB84@yahoo.com', 'Australia', '/files/saraB84.img', '2024-02-26 16:55', '1984-07-10'),

('jchung32', 'Chung, Jane', 'hello1234', '555-4567', 'jchung32@gmail.com', 'South Korea', '/files/jchung32.img', '2024-02-27 08:15', '1998-02-15'),

('mario92', 'Rossi, Mario', 'ciao9876', '555-1111', 'mario92@email.com', 'Italy', '/files/mario92.img', '2024-02-28 11:10', '1992-09-18'),

('nicole39', 'Williams, Nicole', 'nicolepwrd', '555-2222', 'nicole39@gmail.com', 'USA', '/files/nicole39.img', '2024-02-28 17:25', '1981-12-05'),

('rharper75', 'Harper, Ryan', 'qwerty12345', '555-3333', 'rharper75@hotmail.com', 'Canada', '/files/rharper75.img', '2024-02-27 14:50', '1975-05-20'),

('paulo88', 'Silva, Paulo', 'paulopass', '555-4444', 'paulo88@yahoo.com', 'Brazil', '/files/paulo88.img', '2024-02-26 19:05', '1988-11-07'),

('lisa01', 'Johnson, Lisa', 'lisa123!', '555-5555', 'lisa01@gmail.com', 'USA', '/files/lisa01.img', '2024-02-28 20:30', '2001-04-15');

-- Payment

INSERT payment (id\_payment, Method, Price, Commission)

VALUES

(1, 'Через платформу', 7000, 0),

(2, 'Через платформу', 9000, 0),

(3, 'Через платформу', 12000, 0),

(4, 'Перевод по СБП', 8000, 0.05),

(5, 'Перевод по СБП', 12500, 0.07),

(6, 'Перевод по СБП', 13000, 0.03),

(7, 'Перевод по Сберу', 7700, 0.035),

(8, 'Перевод по Сберу', 10000, 0.045),

(9, 'Перевод по Сберу', 13000, 0.04),

(10, 'Перевод по Сберу', 14000, 0.01);

-- Blog

INSERT blog (id\_blog, Title, Content, Date\_of\_publication)

VALUES

(1, 'Как правильно выбрать тренера по фитнесу', 'Советы по выбору тренера, который поможет вам достичь ваших фитнес-целей', '2022-10-01'),

(2, 'Топ 10 рецептов здоровых завтраков', 'Идеи и рецепты для приготовления питательных и вкусных завтраков', '2022-11-15'),

(3, 'Как развить свои навыки публичных выступлений', 'Полезные советы и упражнения для повышения уверенности на публичных выступлениях', '2023-01-05'),

(4, 'Лучшие способы управления своим временем', 'Практические советы и стратегии, чтобы эффективно использовать свое время', '2023-02-20'),

(5, 'Основы программирования: введение в Python', 'Краткое руководство по началу программирования на языке Python', '2023-03-10'),

(6, 'Как стать успешным предпринимателем', 'Советы и лайфхаки для развития бизнеса и достижения успеха в предпринимательской сфере', '2023-04-25'),

(7, 'Путешествие по Италии: лучшие города', 'Гид по самым интересным местам для посещения в Италии', '2023-06-01'),

(8, 'Техники улучшения фотографии', 'Советы по обработке фотографий и созданию качественных изображений', '2023-08-15'),

(9, 'Основы финансового планирования', 'Как правильно управлять своими финансами и планировать будущее', '2023-09-02'),

(10, 'Тайны медитации: путь к внутреннему покою', 'Основы и практика медитации для достижения умиротворенного состояния', '2023-10-20');

-- Content bank

INSERT `content bank` (id\_content\_bank, Access\_rights, Location\_on\_the\_server)

VALUES

(1, 'Общедоступный', '/courses/bank1'),

(2, 'Закрытый', '/courses/bank2'),

(3, 'Частичнодоступный', '/courses/bank3'),

(4, 'Общедоступный', '/courses/bank4'),

(5, 'Закрытый', '/courses/bank5'),

(6, 'Частичнодоступный', '/courses/bank6'),

(7, 'Общедоступный', '/courses/bank7'),

(8, 'Закрытый', '/courses/bank8'),

(9, 'Частичнодоступный', '/courses/bank9'),

(10, 'Общедоступный', '/courses/bank10');

-- Course

INSERT Course (`Short name`, Name, The\_beginning\_of\_the\_course, Duration, Description, Rating, Forum\_id\_forum, Payment\_id\_payment, `Content bank\_id\_content\_bank`)

VALUES

('web101', 'Web Development 101', '2022-10-01', 30, 'An introductory course on web development covering HTML, CSS, and JavaScript.', 4.5, 1, 1, 1),

('python201', 'Python Programming 201', '2022-11-15', 45, 'An intermediate course on Python programming covering advanced topics such as object-oriented programming and data structures.', 4.2, 2, 2, 2),

('java101', 'Introduction to Java Programming', '2023-01-05', 60, 'A beginner-level course on Java programming for those who want to learn the basics of Java.', 4.7, 3, 3, 3),

('dbms202', 'Database Management Systems 202', '2023-02-20', 50, 'An advanced course on database management systems covering topics like normalization, indexing, and transactions.', 4.3, 4, 4, 4),

('ml101', 'Machine Learning 101', '2023-03-10', 30, 'An introductory course on machine learning covering the basics of supervised and unsupervised learning algorithms.', 4.6, 5, 5, 5),

('cybersec201', 'Cybersecurity 201', '2023-04-25', 45, 'An intermediate-level course on cybersecurity covering concepts like network security, ethical hacking, and information protection.', 4.4, 6, 6, 6),

('frontend202', 'Advanced Frontend Development', '2023-06-01', 60, 'An advanced course on frontend development focusing on advanced CSS frameworks, JavaScript libraries, and responsive design.', 4.8, 7, 7, 7),

('backend202', 'Advanced Backend Development', '2023-08-15', 45, 'An advanced course on backend development covering topics such as API development, database integration, and server management.', 4.7, 8, 8, 8),

('ios101', 'iOS App Development 101', '2023-09-02', 30, 'An introductory course on iOS app development covering Swift programming language and basics of iOS app development.', 4.3, 9, 9, 9),

('datasci101', 'Introduction to Data Science', '2023-10-20', 60, 'An introductory course on data science covering data preprocessing, exploratory data analysis, and basic statistical concepts.', 4.5, 10, 10, 10);

-- Notification

INSERT notification (id\_notification, Name, Text, Date\_of\_dispatch)

VALUES

(1, 'Обновление информации', 'В ближайшее время будет проведено обновление информации на нашей платформе. Просим держать вас в курсе изменений.', '2023-05-15 10:30'),

(2, 'Важное обновление', 'Планируется важное обновление платформы. Проходите тестирование, чтобы быть готовыми к изменениям.', '2023-03-10 14:00'),

(3, 'Расширение функционала', 'Мы рады сообщить, что мы расширяем функционал нашей платформы. Скоро будут доступны новые возможности.', '2023-09-20 09:45'),

(4, 'Технические работы', 'В связи с проведением технических работ возможны временные перебои в работе платформы. Просим прощения за неудобства.', '2023-07-05 17:30'),

(5, 'Изменения в интерфейсе', 'Мы обновили интерфейс платформы для улучшения пользовательского опыта. Оцените новые изменения.', '2023-02-28 08:15'),

(6, 'Внимание кражи данных', 'Обратите внимание на безопасность своих данных. Будьте осторожны с личной информацией и паролями.', '2023-10-12 11:20'),

(7, 'Новые обучающие материалы', 'Мы добавили новые обучающие материалы на платформу. Изучайте новые темы и совершенствуйте свои навыки.', '2023-04-02 16:45'),

(8, 'Техническая поддержка', 'Наша техническая поддержка работает круглосуточно. Обратитесь к нам в случае возникновения проблем.', '2023-06-18 13:10'),

(9, 'Обновление программы курса', 'Программа курса была обновлена. Ознакомьтесь с новыми темами и заданиями.', '2023-08-30 10:00'),

(10, 'Получите сертификат', 'Завершите курс и получите сертификат о прохождении. Успейте закончить все задания.', '2023-11-25 15:30');

-- Question

INSERT INTO question (id\_question, Name, Type, Creation\_time, Text, Score\_for\_correct\_execution, The\_maximum\_size\_of\_the\_file\_to\_be\_added, The\_maximum\_number\_of\_files\_to\_add)

VALUES

(1, 'ДЗ-8 вопрос № 1', 'Выбор из списка', '2023-02-15 18:15', 'Расположите процесс преобразование сложной модели в более простую', 12.50, 0, 0),

(2, 'ДЗ-5 вопрос № 1', 'Эссе', '2023-02-16 18:15', 'Напишите реферат на тему "История создания Интернета"', 35.00, 20, 1),

(3, 'КР-2 вопрос № 3', 'Программа', '2023-02-17 18:15', 'Напишите программу на языке программирования Кумир, которая позволяет передвинуть Робота из левого нижнего угла обстановки в правый верхний при условии отсутствия внутренних стен обстановки', 20.00, 0, 0),

(4, 'ДЗ-8 вопрос № 1', 'Выбор из списка', '2023-02-18 18:15', 'Расположите процесс преобразование сложной модели в более простую', 12.50, 0, 0),

(5, 'ДЗ-5 вопрос № 1', 'Эссе', '2023-02-19 18:15', 'Напишите реферат на тему "История создания Интернета"', 35.00, 20, 1),

(6, 'КР-2 вопрос № 3', 'Программа', '2023-02-20 18:15', 'Напишите программу на языке программирования Кумир, которая позволяет передвинуть Робота из левого нижнего угла обстановки в правый верхний при условии отсутствия внутренних стен обстановки', 20.00, 0, 0),

(7, 'ДЗ-8 вопрос № 1', 'Выбор из списка', '2023-02-21 18:15', 'Расположите процесс преобразование сложной модели в более простую', 12.50, 0, 0),

(8, 'ДЗ-5 вопрос № 1', 'Эссе', '2023-02-22 18:15', 'Напишите реферат на тему "История создания Интернета"', 35.00, 20, 1),

(9, 'КР-2 вопрос № 3', 'Программа', '2023-02-23 18:15', 'Напишите программу на языке программирования Кумир, которая позволяет передвинуть Робота из левого нижнего угла обстановки в правый верхний при условии отсутствия внутренних стен обстановки', 20.00, 0, 0),

(10, 'ДЗ-8 вопрос № 1', 'Выбор из списка', '2023-02-24 18:15', 'Расположите процесс преобразование сложной модели в более простую', 12.50, 0, 0);

-- Testing

INSERT testing (id\_test, Start\_date, End\_date, `Content bank\_id\_content\_bank`)

VALUES

(1, '2024-02-15 12:55', '2024-02-19 18:00', 1),

(2, '2024-02-16 12:55', '2024-02-21 12:55', 1),

(3, '2024-02-17 12:55', '2024-02-22 12:55', 2),

(4, '2024-02-18 12:55', '2024-02-23 12:55', 1),

(5, '2024-02-19 12:55', '2024-02-24 18:00', 1),

(6, '2024-02-20 12:55', '2024-02-25 12:55', 3),

(7, '2024-02-21 12:55', '2024-02-26 12:55', 1),

(8, '2024-02-22 12:55', '2024-02-27 12:55', 2),

(9, '2024-02-23 12:55', '2024-02-28 18:00', 1),

(10, '2024-02-24 12:55', '2024-03-01 12:55', 1);

-- Cource element

INSERT `course element` (id\_course\_element, Number, Name, Type, Visibility, Attached\_file, `Content bank\_id\_content\_bank`)

VALUES

(1, 1, 'Посещаемость занятий', 'Отметка посещаемости', 1, NULL, 1),

(2, 2, 'Ссылка на главный архив', 'Ссылка', 1, NULL, 3),

(3, 1, 'Задание на первую практику', 'Задание', 0, '/files/practice.pdf', 2),

(4, 3, 'Пример протокола первой лабы', 'Файл', 1, '/files/protocol.docx', 3),

(5, 2, 'Анкета о качестве обучения', 'Анкетирование', 0, NULL, 2),

(6, 1, 'Видеолекции по предмету', 'Папка', 1, '/files/lecture', 1),

(7, 3, 'Тест по теме', 'Тест', 1, NULL, 4),

(8, 2, 'Результаты контрольной работы', 'Файл', 0, '/files/exam\_results.docx', 1),

(9, 4, 'Итоговое задание', 'Задание', 1, NULL, 3),

(10, 3, 'Форум обсуждений', 'Форум', 1, NULL, 2);

-- Testing to Question

INSERT `testing to question` (Question\_id\_question, Testing\_id\_test, Number)

VALUES

(1, 1, 2),

(2, 1, 1),

(3, 2, 1),

(3, 4, 1),

(4, 3, 2),

(5, 3, 1),

(6, 2, 1),

(6, 4, 1),

(7, 1, 2),

(8, 1, 1);

-- User to course

INSERT `user to course` (User\_Login, `Course\_Short name`, Role)

VALUES

('3pknai', 'python201', 'Преподаватель'),

('3pknai', 'frontend202', 'Ученик'),

('jdoe78', 'backend202', 'Ученик'),

('jdoe78', 'python201', 'Ученик'),

('lisa01', 'frontend202', 'Преподаватель'),

('jdoe78', 'frontend202', 'Преподаватель'),

('lisa01', 'python201', 'Преподаватель'),

('mario92', 'backend202', 'Ученик'),

('mario92', 'python201', 'Преподаватель'),

('mario92', 'frontend202', 'Ученик');

-- User to blog

INSERT `user to blog` (User\_Login, Blog\_id\_blog, Privilege)

VALUES

('3pknai', 2, 'Создатель'),

('3pknai', 1, 'Редактор'),

('lisa01', 2, 'Редактор'),

('lisa01', 3, 'Читатель'),

('mario92', 1, 'Читатель'),

('mario92', 3, 'Читатель'),

('mario92', 2, 'Читатель'),

('alex87', 1, 'Редактор'),

('alex87', 3, 'Редактор'),

('rharper75', 1, 'Читатель');

-- User to notification

INSERT `user to notification` (User\_Login, Notification\_id\_notification, `Read`)

VALUES

('lisa01', 1, 0),

('3pknai', 1, 0),

('lisa01', 2, 1),

('mario92', 2, 0),

('mario92', 1, 1),

('alex87', 2, 0),

('alex87', 3, 1),

('rharper75', 1, 0),

('rharper75', 2, 1),

('rharper75', 3, 1);

-- User to Question

INSERT `user to question` (Question\_id\_question, User\_Login, The\_score\_received)

VALUES

(1, 'lisa01', 12.50),

(3, 'lisa01', 20.00),

(1, '3pknai', 12.50),

(2, '3pknai', 35.00),

(3, '3pknai', 19.00),

(4, 'alex87', 25.00),

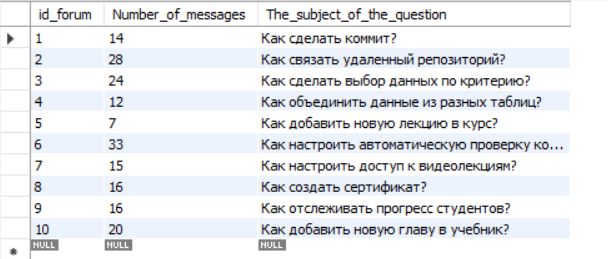
(5, 'alex87', 17.50),

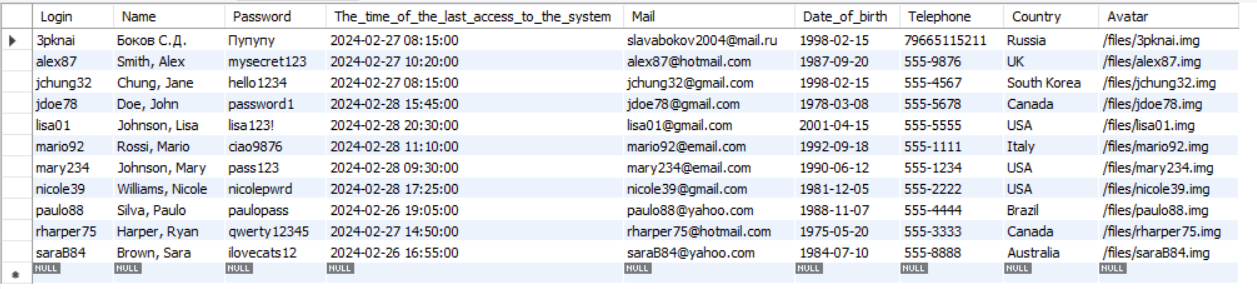
(1, 'rharper75', 13.50),

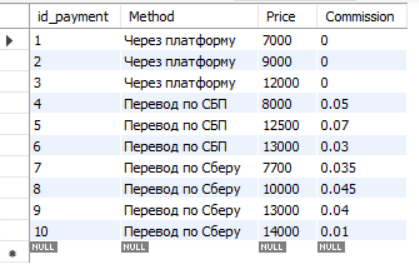
(2, 'rharper75', 30.00),

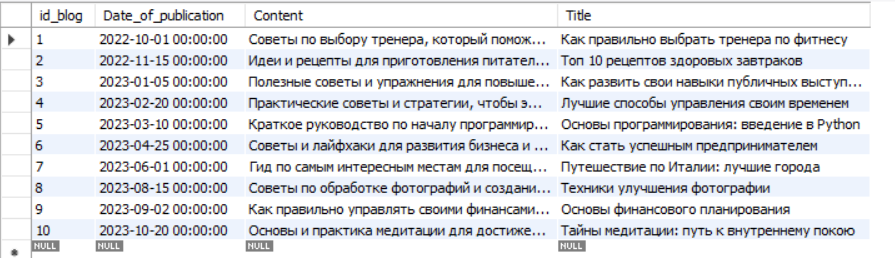
(3, 'rharper75', 18.00);

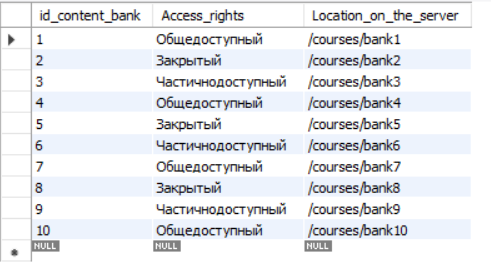
**Скриншоты полученных таблиц в интерфейсе MySQL**:

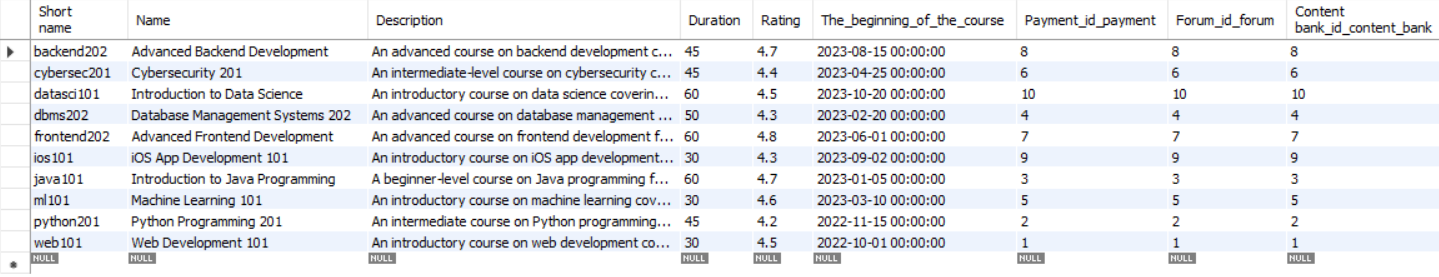


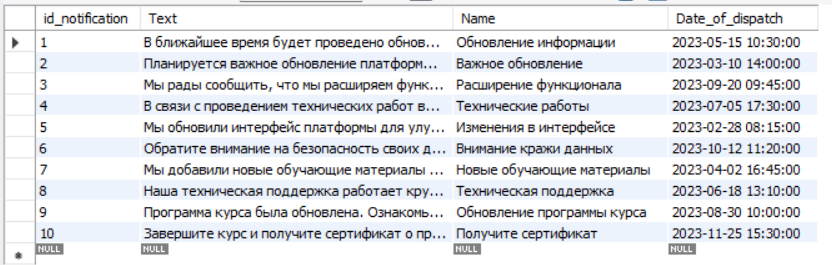


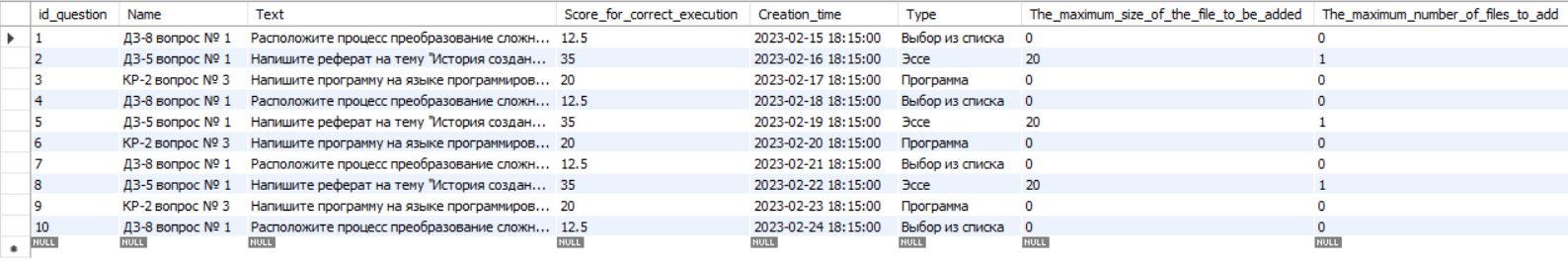


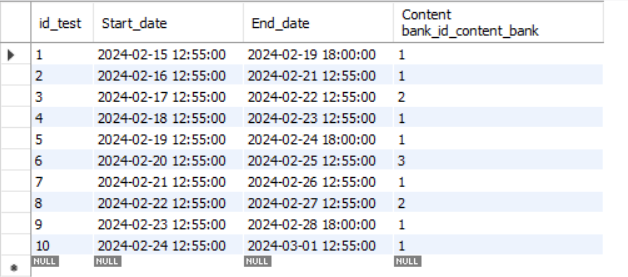


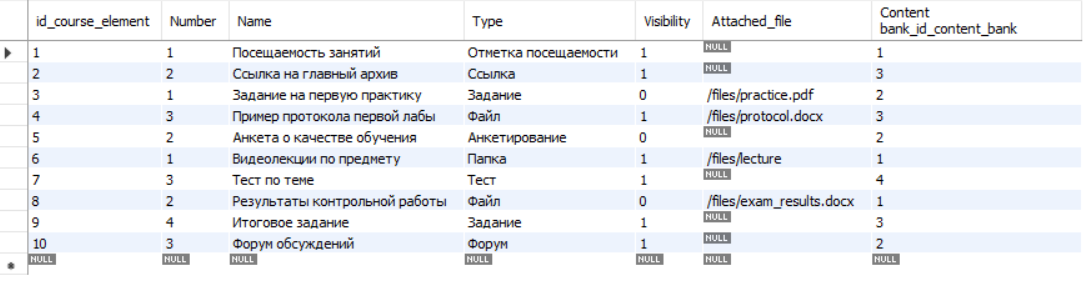


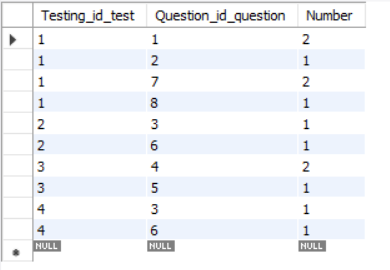


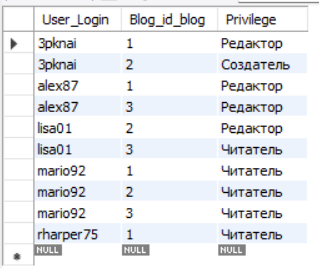


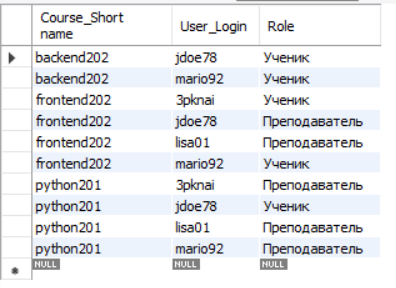


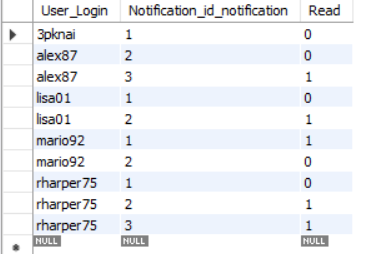


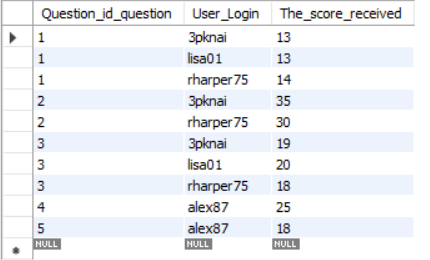






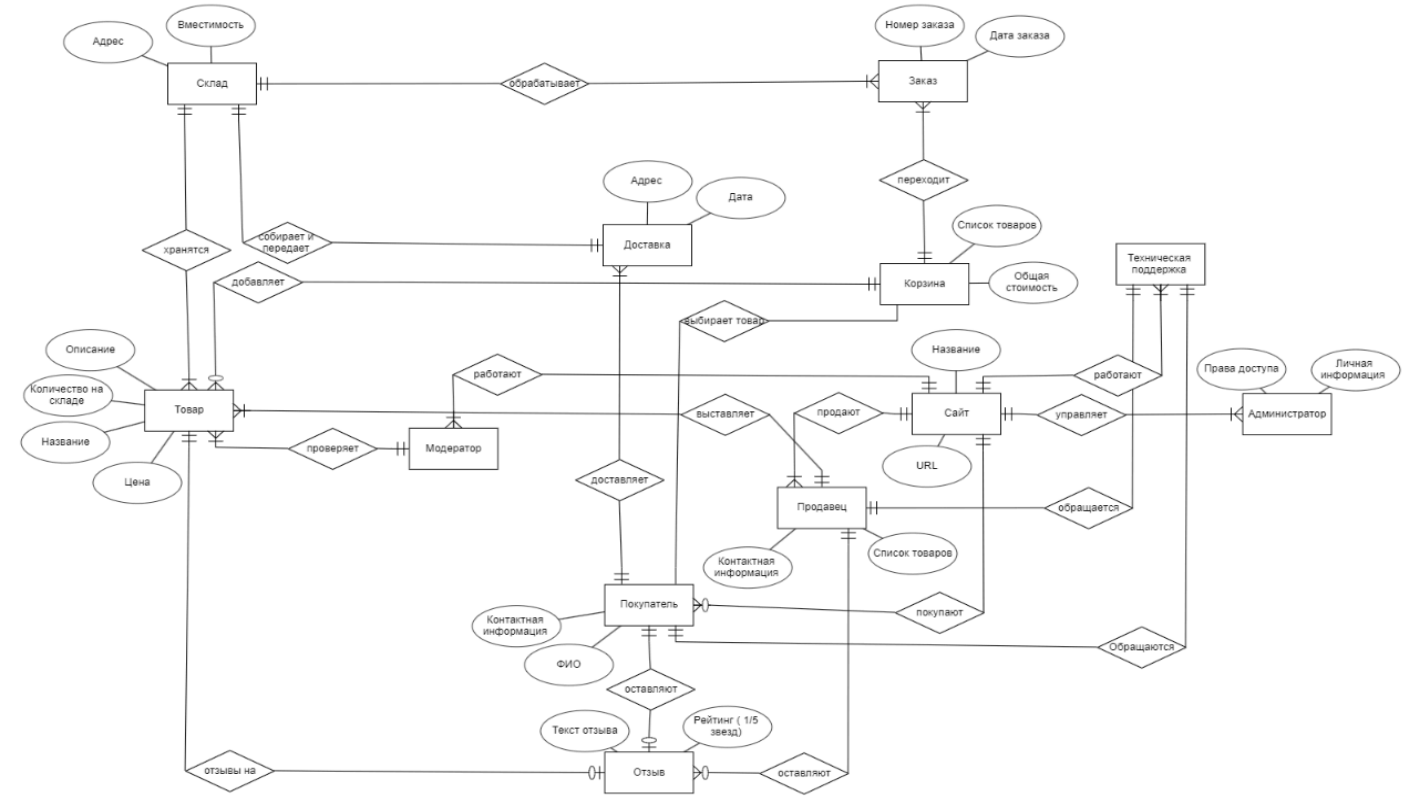




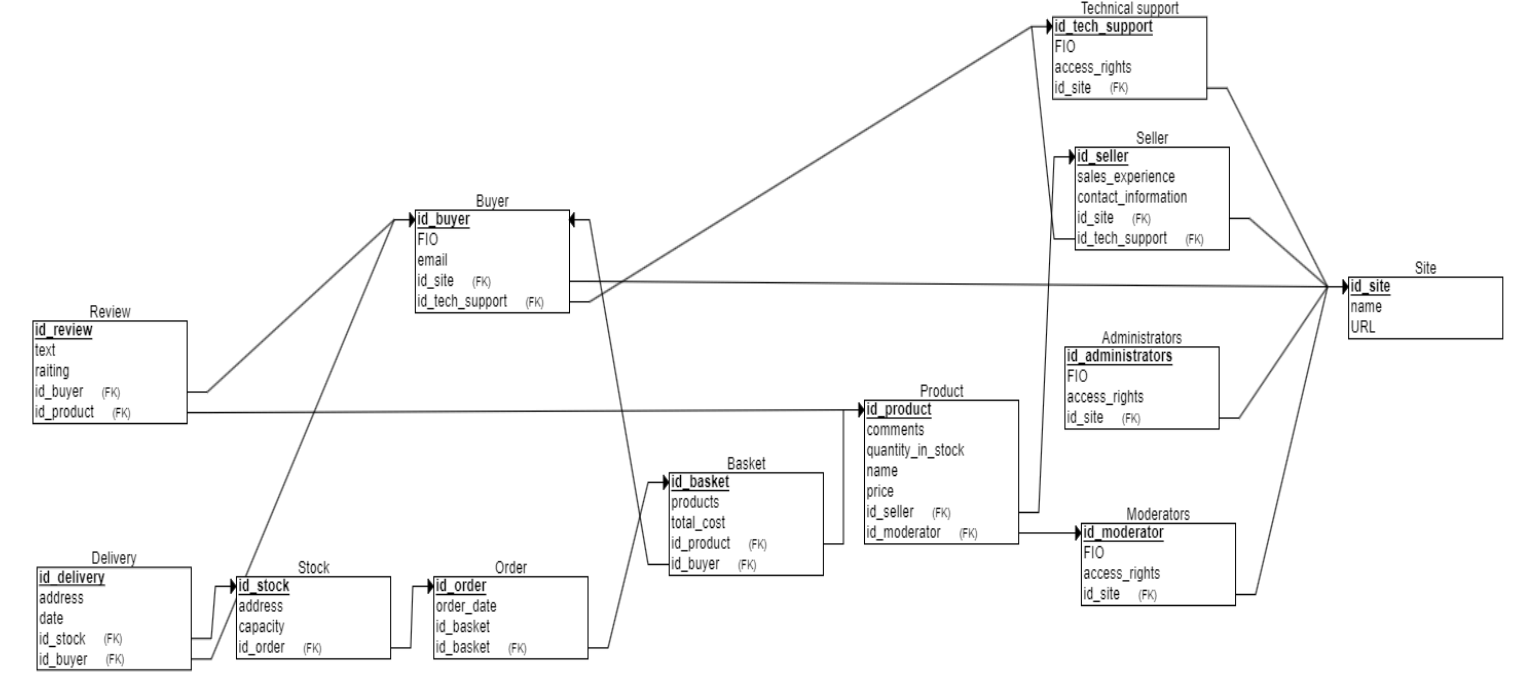


Реализация БД одногруппника

**ER-диаграмма:**



**Реляционная схема:**



**SQL скрипт для создания этой базы данных:**

CREATE database `Knitted Toy Store`;

use `Knitted Toy Store`;

CREATE table `site`(

id\_site int not null auto\_increment,

name varchar(26) not null,

url varchar(80) not null,

primary key (id\_site)

);

CREATE table `administrators`(

id\_administrators int not null auto\_increment,

FIO varchar(50) not null,

access\_rights varchar(30) not null,

id\_site int not null,

foreign key (id\_site) references site (id\_site),

primary key (id\_administrators)

);

CREATE table `moderators`(

id\_moderator int not null auto\_increment,

FIO varchar(50) not null,

access\_rights varchar(30) not null,

id\_site int not null,

foreign key (id\_site) references site (id\_site),

primary key (id\_moderator)

);

CREATE table `technical support`(

id\_tech\_support int not null auto\_increment,

FIO varchar(50) not null,

access\_rights varchar(30) not null,

id\_site int not null,

foreign key (id\_site) references site (id\_site),

primary key (id\_tech\_support)

);

CREATE table `seller`(

id\_seller int not null auto\_increment,

contact\_information varchar(80) not null,

id\_site int not null,

id\_tech\_support int not null,

foreign key (id\_site) references site (id\_site),

foreign key (id\_tech\_support) references `technical support` (id\_tech\_support),

primary key (id\_seller)

);

CREATE table `product`(

id\_product int not null auto\_increment,

comments varchar(80) not null,

quantity\_in\_stock int not null,

name varchar(26) not null,

id\_seller int not null,

id\_moderator int not null,

foreign key (id\_seller) references seller (id\_seller),

foreign key (id\_moderator) references `moderators` (id\_moderator),

primary key (id\_product)

);

CREATE table `buyer`(

id\_buyer int not null auto\_increment,

FIO varchar(50) not null,

email varchar(30) not null,

id\_site int not null,

id\_tech\_support int not null,

foreign key (id\_site) references site (id\_site),

foreign key (id\_tech\_support) references `technical support` (id\_tech\_support),

primary key (id\_buyer)

);

CREATE table `review`(

id\_review int not null auto\_increment,

text text,

rating float,

id\_buyer int not null,

id\_product int not null,

foreign key (id\_buyer) references buyer (id\_buyer),

foreign key (id\_product) references product (id\_product),

primary key (id\_review)

);

CREATE table `basket`(

id\_basket int not null auto\_increment,

capacity int not null,

id\_buyer int not null,

foreign key (id\_buyer) references buyer (id\_buyer),

primary key (id\_basket)

);

CREATE table `basket to product`(

id\_basket int not null,

id\_product int not null,

foreign key (id\_product) references product (id\_product),

foreign key (id\_basket) references basket (id\_basket),

primary key (id\_basket, id\_product)

);

CREATE table `order`(

id\_order int not null auto\_increment,

order\_date datetime not null,

id\_basket int not null,

foreign key (id\_basket) references basket (id\_basket),

primary key (id\_order)

);

CREATE table `stock`(

id\_stock int not null auto\_increment,

address varchar(100) not null,

capacity int not null,

id\_order int not null,

foreign key (id\_order) references `order` (id\_order),

primary key (id\_stock)

);

CREATE table `delivery`(

id\_delivery int not null auto\_increment,

address varchar(100) not null,

date date,

id\_stock int not null,

id\_buyer int not null,

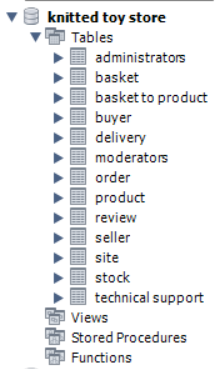
foreign key (id\_stock) references stock (id\_stock),

foreign key (id\_buyer) references buyer (id\_buyer),

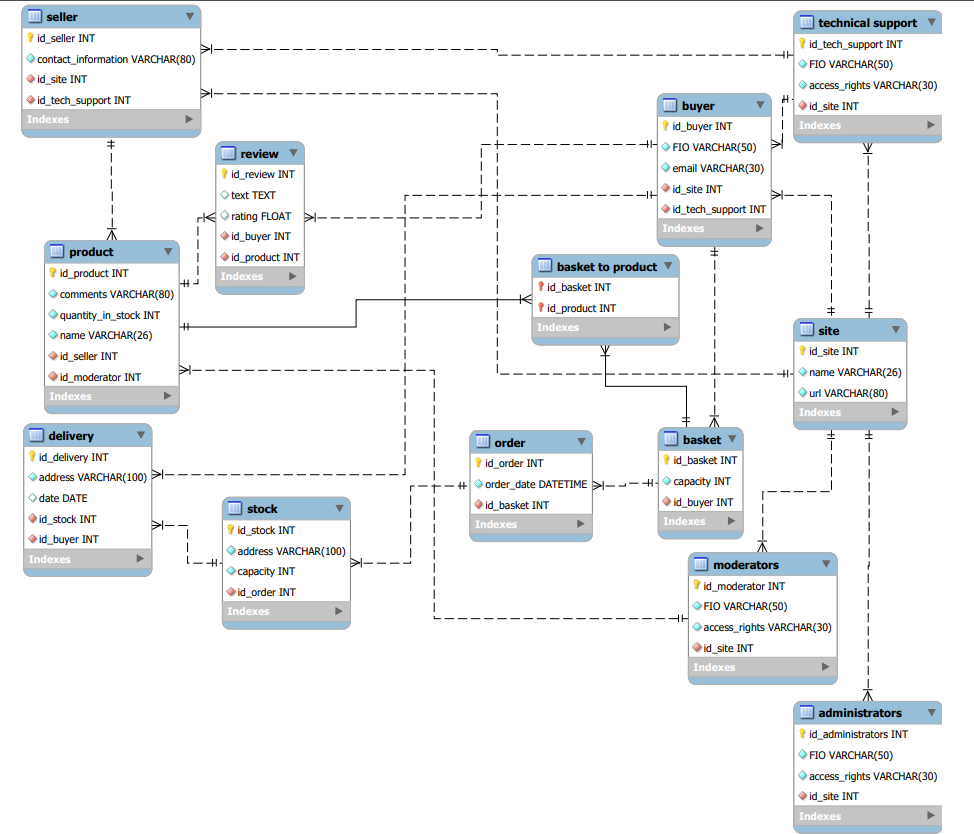
primary key (id\_delivery)

);

**Был сгенерирован каркас базы данных:**



**Полученная диаграмма:**



**SQL скрипт для заполнения таблиц данными:**

-- Site

INSERT site(id\_site, name, url)

VALUES

(1, "distriction", "distriction.ru");

-- Technical support

INSERT `technical support`(id\_tech\_support, FIO, access\_rights, id\_site)

VALUES

(1, "Матохин Илья Георгиевич", "Полный", 1);

-- Administrators

INSERT administrators(id\_administrators, FIO, access\_rights, id\_site)

VALUES

(1, "Зенин Максим Александрович", "Полный", 1);

-- Moderators

INSERT moderators(id\_moderator, FIO, access\_rights, id\_site)

VALUES

(1, "Боков Святослав Дмитриевич", "Частичный", 1);

-- Seller

INSERT seller(id\_seller, contact\_information, id\_site, id\_tech\_support)

VALUES

(1, "superpetya.ru", 1, 1);

-- Product

INSERT product(id\_product, comments, quantity\_in\_stock, name, id\_seller, id\_moderator)

VALUES

(1, "Лучшая мягкая игрушка в мире!", 7, "Мышь", 1, 1),

(2, "Лучшая мягкая игрушка на свете!", 4, "Дракон", 1, 1);

-- Buyer

INSERT buyer(id\_buyer, FIO, email, id\_site, id\_tech\_support)

VALUES

(1, "Молчанова Лиана Евгеньевна", "lihimiko.mail.ru", 1, 1);

-- Review

INSERT review(id\_review, text, rating, id\_buyer, id\_product)

VALUES

(1, "Товар пришёл в ценности", 5, 1, 1);

-- Basket

INSERT basket(id\_basket, capacity, id\_buyer)

VALUES

(1, 5, 1);

-- Basket to product

INSERT `basket to product`(id\_basket, id\_product)

VALUES

(1, 1),

(1, 2);

-- Order

INSERT `order`(id\_order, order\_date, id\_basket)

VALUES

(1, "2024-02-15 12:00", 1);

-- Stock

INSERT stock(id\_stock, address, capacity, id\_order)

VALUES

(1, "ул. Н. Отрады 10а", 500, 1);

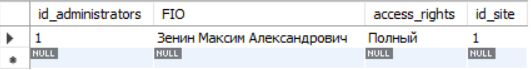
-- Delivery

INSERT delivery(id\_delivery, address, date, id\_stock, id\_buyer)

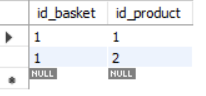
VALUES

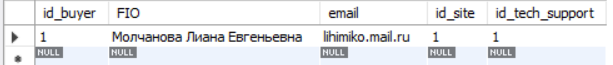
(1, "ул. Н. Отрады 22 173", "2024-02-16", 1, 1);

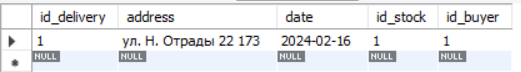
**Скриншоты таблиц с данными:**

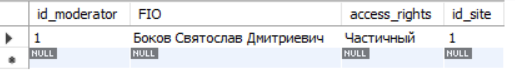


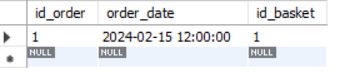


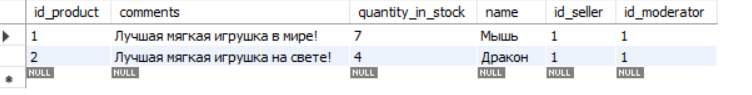




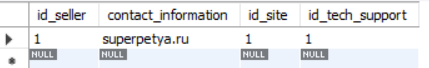


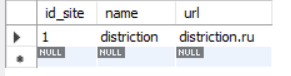


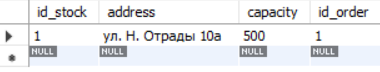


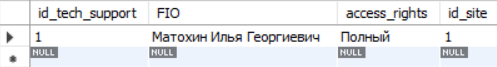












**SQL скрипт для изменения таблиц:**

-- Добавление поля количество в связь между корзиной и товаром

ALTER TABLE `basket to product` ADD count INT NOT NULL DEFAULT 1;

-- Изменение столбца таблицы (увеличение количества символов)

ALTER TABLE buyer modify email varchar(50);

ALTER TABLE moderators ADD id\_administrators int;

-- Добавление внешнего ключа

ALTER TABLE moderators ADD FOREIGN KEY (id\_administrators) REFERENCES administrators(id\_administrators);

-- Удаление внешнего ключа

ALTER TABLE moderators DROP constraint moderators\_ibfk\_2;

-- Удаление первичного ключа

ALTER TABLE delivery DROP id\_delivery;

-- Добавление первичного ключа

ALTER TABLE delivery ADD PRIMARY KEY (address);

-- Добавление ограничения уникальности

ALTER TABLE buyer ADD CONSTRAINT email\_unq UNIQUE(email);

-- Переименование столбца таблицы

ALTER TABLE administrators RENAME COLUMN id\_administrators TO id\_administrator;

-- Переименование таблицы

ALTER TABLE `technical support` RENAME TO tech\_support;

**SQL скрипт для удаления базы данных и таблиц**

drop table review;

drop table `basket to product`;

drop table delivery;

drop table stock;

drop table `order`;

drop table basket;

drop table buyer;

drop table product;

drop table seller;

drop table tech\_support;

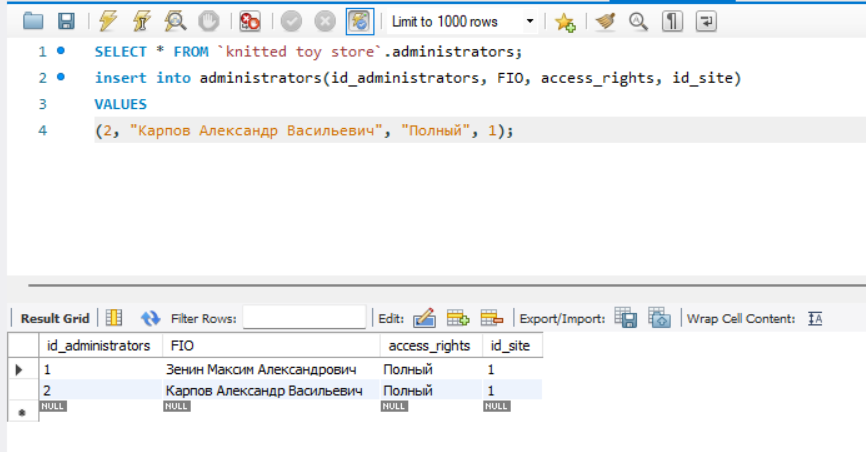
drop table administrators;

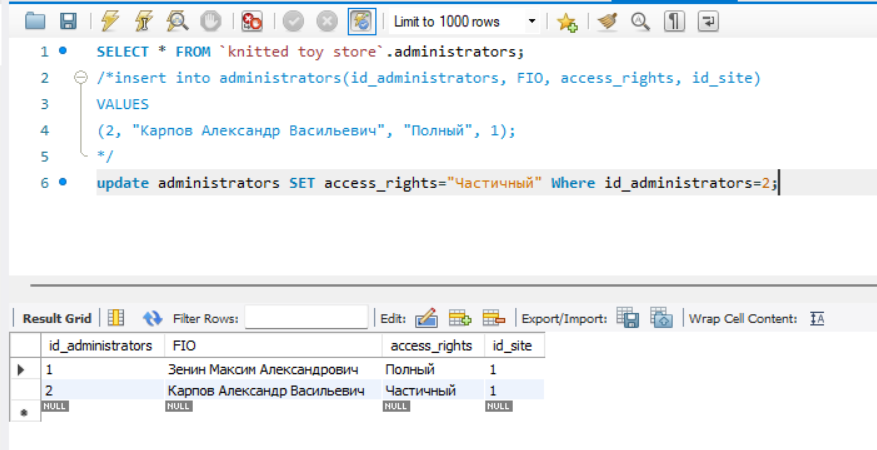
drop table moderators;

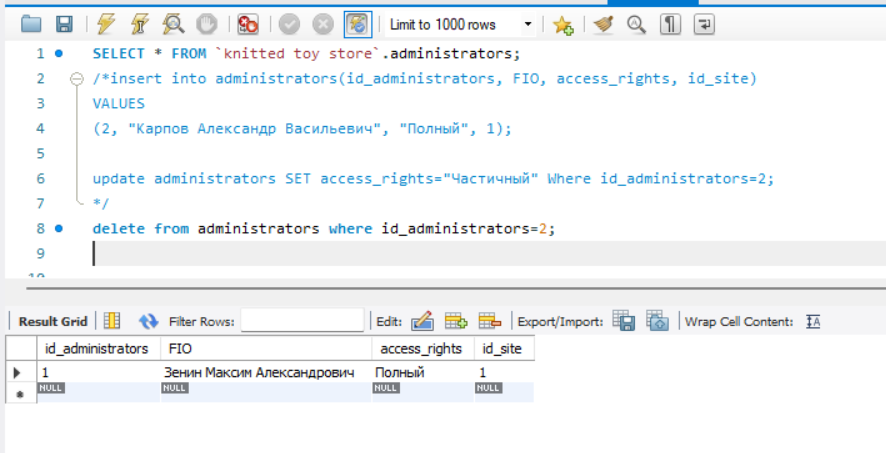
drop table site;

drop database `knitted toy store`;

**Запросы, которые были написаны при отчёте**







**Модификация:**

