

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФ. М.А. БОНЧ-БРУЕВИЧА»
(СПбГУТ)**

**АРХАНГЕЛЬСКИЙ КОЛЛЕДЖ ТЕЛЕКОММУНИКАЦИЙ (ФИЛИАЛ) СПбГУТ
(АКТ (ф) СПбГУТ)**

**Составил
Ю.С. Маломан**

УП.11 ТЕХНОЛОГИЯ РАЗРАБОТКИ И ЗАЩИТЫ БАЗ ДАННЫХ

**Сборник описаний практических работ
(часть 1)**

Архангельск 2024

Маломан, Ю.С. УП.11 Технология разработки и защиты баз данных. Сборник описаний практических работ. - Архангельск: АКТ (ф) СПбГУТ, 2024.

Сборник описаний практических работ по УП.11 Технология разработки и защиты баз данных. Нумерация таблиц дана в пределах одной работы. Сборник предназначен для студентов очной формы обучения по специальности: 09.02.07 Информационные системы и программирование

Рассмотрено и одобрено на заседании цикловой комиссии Информационных технологий и математических дисциплин Архангельского колледжа телекоммуникаций (филиал) СПбГУТ.

Содержание

- Практическая работа №1 Разработка физической модели данных БД
- Практическая работа №2 Проведение прямого инжиниринга модели данных в БД MySQL
- Практическая работа №3 Создание однотабличных SQL-запросов на выборку данных в MySQL
- Практическая работа №4 Создание многотабличных SQL-запросов на выборку данных в MySQL
- Практическая работа №5 Создание SQL-запросов на модификацию данных в MySQL
- Практическая работа №6 Применение встроенных функций MySQL
- Практическая работа №7 Осуществление полнотекстового поиска в БД MySQL
- Практическая работа №8 Применение транзакций в MySQL
- Практическая работа №9 Применение планировщика событий
- Практическая работа №10 Создание SQL-запросов на модификацию схемы БД в MySQL
- Практическая работа №11 Проведение обратного инжиниринга модели данных из БД MySQL
- Практическая работа №12 Создание представлений в MySQL
- Практическая работа №13 Создание функций пользователя в MySQL
- Практическая работа №14 Создание хранимых процедур в MySQL
- Практическая работа №15 Создание AFTER-триггеров в MySQL
- Практическая работа №16 Создание BEFORE-триггеров в MySQL
- Практическая работа №17 Разграничение прав доступа пользователя к БД MySQL
- Практическая работа №18 Резервное копирование и восстановление БД MySQL
- Практическая работа №19 Разработка приложения для авторизации пользователей MySQL
- Практическая работа №20 Разработка приложения для выполнения запросов на выборку данных из БД MySQL
- Практическая работа №21 Разработка приложения для выполнения запросов на модификацию данных в БД MySQL

Практическая работа №1

Разработка физической модели данных БД

1 Цель работы

1.1 Научиться создавать физическую модель данных БД MySQL, используя CASE-средство MySQL Workbench.

2 Литература

2.1 Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – 4-е изд., перераб. и доп. – Москва: ИД| ФОРУМ: НИЦ Инфра-М, 2012. – с.29-33.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Описание предметной области «Книжный магазин»:

Книжный интернет-магазин принимает заказы на доставку книг в соответствии с имеющимся ассортиментом и прейскурантом.

У каждой книги указывается ее наименование и только один автор. При этом считать, что у автора имя и фамилия совместно уникальны, а страна по умолчанию — Россия. Также в информации о книгах могут отображаться цена (меньше 10000 рублей), масса (указывается в килограммах, меньше 10 кг), количество страниц и год выпуска (не позже текущего), жанр (проза, поэзия, другое, по умолчанию — проза).

Каждая книга может присутствовать в нескольких заказах. Заказ оформляется только на одну или более книг в количестве от 1 до 100 (по умолчанию в количестве книг заказа указывается один экземпляр), после оформления ему присваивается код, который сообщается покупателю, и ставится текущая дата как дата заказа.

Каждый покупатель может сделать любое количество заказов. Покупатель при регистрации указывает уникальный логин, имя, фамилию и адрес, а также может указать телефон, после регистрации ему присваивается код покупателя.

5.1 Создание БД и таблиц

Создать в MySQL Workbench EER-модель. Назвать созданную схему «Market».

Добавить на рабочую область все таблицы БД (описаны далее в таблицах 1-5), кроме таблицы Состав.

В каждую таблицу добавить все атрибуты кроме атрибутов внешних ключей.

Отметить первичные ключи (отмечаются флажком в столбце PK).

Все названия указывать на английском языке без пробелов (вместо пробелов — нижнее подчеркивание для snake_case или заглавная буква для CamelCase).

5.2 Настройка типов данных и обязательности столбцов

Настроить типы данных столбцов. При настройке использовать сведения, указанные в таблицах 1-4.

У числовых столбцов, в которых не должно быть отрицательных значений, проставить флажок UN (беззнаковый).

Указать обязательность заполнения, поставив в столбце NN флажок. Необязательны только те столбцы, про которые это сказано в таблицах 1-4.

5.3 Создание связей между таблицами

Настроить связи между таблицами с учетом следующих правил:

— у каждой книги указывается только один автор (добавить неидентифицирующую связь 1:M);

— каждый покупатель может сделать несколько заказов (добавить неидентифицирующую связь 1:M).

— в каждом заказе может быть несколько книг в разном количестве (добавить связь M:M, появившуюся связующую таблицу для связи M:M назвать Состав и добавить в нее недостающий столбец согласно описанию в таблице 5). Настроить типы данных и обязательность столбцов.

В настройках Foreign Keys таблиц обеспечить каскадное обновление записей, при удалении – бездействие.

Изменить названия столбцов внешних ключей, чтобы они совпадали с названиями связанных столбцов первичных ключей.

5.4 Указать для каждой таблицы значения по умолчанию. Первичные ключи, не являющиеся внешними ключами, должны быть автоинкрементными для каждой таблицы БД.

5.5 Указать для каждой таблицы уникальность полей (фамилия и имя автора совместно уникальны, настраиваются через вкладку Indexes таблиц), создав соответствующие индексы.

Таблица 1 — Поля таблицы «Авторы»

Поле	Тип данных	Примечание	Значение по умолчанию
Код автора	INT	Первичный ключ	
Фамилия	VARCHAR	Размер – 50	
Имя	VARCHAR	Размер – 50	
Страна	VARCHAR	Размер – 30	'Россия'

Таблица 2 — Поля таблицы «Книги»

Поле	Тип данных	Примечание	Значение по умолчанию
Код книги	INT	Первичный ключ	
Код автора	INT	Внешний ключ	
Название	VARCHAR	Размер – 50	
Жанр	ENUM	Допустимые значения типа: 'проза', 'поэзия', 'другое'	'проза'
Цена	DECIMAL	От 0 до 9999.99	0
Масса	DECIMAL	От 0.000 до 9.999 кг	0
Страницы	SMALLINT		0
Год выпуска	YEAR	необязательное	

Таблица 3 — Поля таблицы «Заказчики»

Поле	Тип данных	Примечание	Значение по умолчанию
Код заказчика	INT	Первичный ключ	
Логин	VARCHAR	Размер – 20, уникальное	
Фамилия	VARCHAR	Размер – 50	
Имя	VARCHAR	Размер – 50	
Адрес	VARCHAR	Размер – 100	
Телефон	VARCHAR	Размер – 20, необязательное	

Таблица 4 — Поля таблицы «Заказы»

Поле	Тип данных	Примечание	Значение по умолчанию
Номер заказа	INT	Первичный ключ	
Код заказчика	INT	Внешний ключ	
Дата и время заказа	DATETIME		CURRENT_TIMESTAMP

Таблица 5 — Поля таблицы «Состав»

Поле	Тип данных	Примечание	Значение по умолчанию
Номер заказа	INT	Первичный ключ, внешний ключ	
Код книги	INT	Первичный ключ, внешний ключ	
Количество	TINYINT		1

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего применяется MySQL Workbench?
- 8.2 Как создать EER-модель в MySQL Workbench?
- 8.3 Из каких элементов состоит EER-модель?
- 8.4 Как создаются индексы в MySQL Workbench?
- 8.5 Какие виды индексов можно создать в таблицах MySQL?

Практическая работа №2

Проведение прямого инжиниринга модели данных в БД MySQL

1 Цель работы

1.1 Научиться проводить прямой инжиниринг модели данных в БД MySQL, используя CASE-средство MySQL Workbench.

2 Литература

2.1 Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – 4-е изд., перераб. и доп. – Москва: ИД| ФОРУМ: НИЦ Инфра-М, 2012. – с.101-123.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Запуск локального сервера

5.1.1 Разархивировать папку с сервером OpenServer в папку C:\Temp\[название группы]\OpenServer.

5.1.2 Открыть папку «OpenServer» и запустить на выполнение файл «Open Server.exe».

5.1.3 В панели задач выбрать пиктограмму сервера (пиктограмма: красный флаг) — ПКМ — Запустить.

5.1.4 В панели задач выбрать пиктограмму запущенного сервера (пиктограмма: зеленый флаг) — ПКМ — Дополнительно — PhpMyAdmin или PHPAdminer. Откроется страница веб-интерфейса для управления СУБД MySQL.

Логин: root, пароль: root

5.2 Провести прямой инжиниринг созданной на практической работе №1 EER-модели из MySQL Workbench в локальную СУБД MySQL. Для этого:

- открыть модель, выбрать в пункте меню Database подпункт Forward Engine, указать настройки подключения к локальному серверу MySQL:

hostname: 127.0.0.1, port: 3306, username: root, password: root;

- в открывшейся форме выбрать пункт Add SHOW WARNINGS.

- нажимать Next до тех пор, пока не появится скрипт на создание таблиц.

- внести изменения в скрипт, добавив в скрипт создания таблицы Состав после описания столбцов проверочное ограничение:

CONSTRAINT название CHECK (количество > 0 AND количество <= 100),

- сохранить в отдельном файле сгенерированный скрипт, нажав Save to file,

- нажимать Next до тех пор, пока не появится Forward Engineer Progress.

Если все корректно, на экран должен быть выведен текст:

Forward Engineer Finished Successfully

Если в логе есть ошибки, исправить их и повторно провести прямой инжиниринг.

5.3 Подключиться к БД через MySQL Workbench. Для этого выбрать в пункте меню Database подпункт Connect to Database и подключиться сначала к созданной на удаленном сервере БД (указывается в поле Default Schema или может быть выбрана после подключения во вкладке Navigator – Schemas – Обновить), затем – к БД, созданной на локальном сервере.

5.4 Заполнить таблицы БД на локальном сервере данными.

Для этого в правой панели Object Browser выбрать БД. В раскрывающемся списке выбрать таблицу. В контекстном меню таблиц пункт Select Rows открывает таблицу в режиме чтения и редактирования. Для принятия изменений нажимать на кнопку Apply (в правом нижнем углу).

В таблицу «авторы» добавить 5 записей, в «книги» — 7, в «заказчики» — 3, в «заказы» — 5, в «состав» — 10.

При вводе использовать значения по умолчанию, автоинкрементные значения.

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench.

6.2 Выполнить задания из п.5.

6.5 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Для чего применяется MySQL Workbench?

8.2 Что такое «прямой инжиниринг»?

8.3 Как провести прямой инжиниринг, используя MySQL Workbench?

8.4 Что включает в себя скрипт, который генерируется в ходе прямого инжиниринга?

Практическая работа №3

Создание однотабличных SQL-запросов на выборку данных в MySQL

1 Цель работы

1.1 Научиться создавать простые однотабличные SQL-запросы на выборку в MySQL.

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.631-633.

2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-229.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Получить все данные обо всех авторах (подсказка: *).

5.2 Получить выборку данных, состоящую из столбцов Фамилия и Имя для первых трех авторов (подсказка: LIMIT)

5.3 Получить выборку данных из одного столбца, содержащего страны происхождения авторов без дубликатов, упорядоченные по алфавиту (подсказка: DISTINCT, ORDER BY).

5.4 Получить выборку данных о заказчиках, состоящую из четырех столбцов:

- 1 столбец – идентификатор книги;
- 2 столбец – название книги.
- 3 столбец – скидка (должен содержать значение 5% во всех записях);
- 4 столбец – цена со скидкой.

5.5 Получить выборку данных, содержащую информацию о книгах с указанием категории цены (дешевые (до 100), средние (от 100 до 1000), дорогие (от 1000)). Отсортировать по убыванию цены (подсказка: CASE с условиями).

5.6 Получить выборку данных, содержащую Логин, Фамилию, Имя и Телефон для всех заказчиков, у которых указан телефон (подсказка: WHERE, IS NOT NULL).

5.7 Получить выборку данных, содержащую наименования книг, которые содержат подстроку “компьютер” (подсказка: LIKE).

5.8 Получить выборку данных, содержащую минимальную, максимальную и среднюю стоимость книг (подсказка: MIN, MAX, AVG).

5.9 Получить выборку данных, содержащую идентификатор автора, уникальные названия книг и количество написанных автором под каждым названием книг (подсказка: COUNT).

5.10 Получить выборку данных, содержащую названия стран и количество авторов в них для тех стран, к которым приписано больше одного автора (подсказка: HAVING).

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench, подключиться к БД Market.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какова структура инструкции SELECT?

8.2 Что указывается после ключевого слова SELECT?

8.3 Что указывается после ключевого слова FROM?

8.4 Что указывается после ключевого слова WHERE?

8.5 В чем отличие между разделами WHERE и HAVING?

8.6 Что указывается после ключевого слова GROUP BY?

8.7 Применение каких функций языка SQL требует группировки столбцов?

8.8 Что указывается после ключевого слова ORDER BY?

8.9 Как создать вычисляемое поле в запросе?

Практическая работа №4

Создание многотабличных SQL-запросов на выборку данных в MySQL

1 Цель работы

- 1.1 Научиться создавать многотабличные SQL-запросы на выборку в MySQL;
- 1.2 Научиться создавать многотабличные SQL-запросы, включающие подзапросы;
- 1.3 Закрепить навык создания запросов на выборку.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.633-637.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-229.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Получить выборку данных содержащую следующие данные о книгах, написанных русскими авторами: фамилия и имя автора, наименование книги, ее цена (подсказка: INNER JOIN).
- 5.2 Получить выборку данных, содержащую фамилию и имя автора и количество имеющихся книг каждого автора (подсказка: COUNT, GROUP BY, LEFT/RIGHT JOIN).
- 5.3 Вывести имя и фамилию всех авторов, которые не написали книг, название которых содержало бы слова “linux” или “windows” (подсказка: подзапрос, NOT IN, LIKE).
- 5.4 Для каждого заказчика вывести его логин, количество сделанных заказов, количество купленных книг и потраченную на заказы книг сумму (подсказка: COUNT, SUM, GROUP BY).
- 5.5 Получить выборку данных, содержащую логины заказчиков, общее количество сделанных ими заказов и купленных ими книг, которые приобрели больше десяти книг (подсказка: HAVING).

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что указывается после ключевого слова FROM?
- 8.2 Каков синтаксис инструкции объединения двух таблиц?
- 8.3 Какие виды объединения таблиц существуют?
- 8.4 Чем отличается LEFT OUTER JOIN от RIGHT OUTER JOIN?
- 8.5 Что такое подзапрос?
- 8.6 Для чего применяется ключевое слово UNION?

Практическая работа №5

Создание SQL-запросов на изменение данных в MySQL

1 Цель работы

1.1 Научиться создавать SQL-запросы на вставку, изменение и удаление данных в MySQL;

1.2 Изучить на практике особенности синтаксиса DML-команд языка MySQL (INSERT, UPDATE, DELETE).

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.626-630.

2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-229.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Заполнить таблицу tempBooks выборкой данных, содержащей фамилию и имя автора, наименование и цену книги для всех книг. Для этого предварительно создать указанную таблицу (типы данных выбрать согласно типам данных исходных таблиц Авторы и Книги) и выполнить следующую команду на вставку:

```
INSERT INTO tempBooks  
SELECT ...
```

5.2 Удалить из таблицы tempBooks все книги, наименование которых содержит слово 'компьютер'.

5.3 Изменить в таблице tempBooks цену книг Пушкина, увеличив ее вдвое, Иванова — уменьшив на 50 рублей, у остальных оставить без изменения. При выполнении использовать один UPDATE и CASE с параметром.

5.4 Удалить из таблицы tempBooks все записи.

5.5 Увеличить на 100 рублей цену всех книг русских авторов.

5.6 Удалить из таблицы Заказчики тех заказчиков, которые не сделали ни одного заказа.

5.7 Добавить в таблицу Авторы нового автора, используя ключевое слово REPLACE (аналог INSERT, но при совпадении уникального индекса или первичного ключа старая запись с совпадающими данными будет удалена).

5.8 Добавить в таблицу Авторы нового автора, используя ON DUPLICATE KEY: если указанные фамилия и имя уже существуют, изменять значение страны на то же, что указано в INSERT.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Каковы причины распространенных ошибок, связанных с выполнением инструкции INSERT INTO?
- 8.2 После какого ключевого слова в запросе на обновление записывается обновляемое поле?
- 8.3 Сколько полей можно обновлять в одном запросе на обновление?
- 8.4 Сколько таблиц можно обновлять в одном запросе на обновление?
- 8.5 Как с помощью инструкции DELETE удалить записи из связанных таблиц?
- 8.6 Какими должны быть две таблицы, чтобы осуществить групповой ввод записей из одной таблицы в другую?

Практическая работа №6

Применение встроенных функций MySQL

1 Цель работы

- 1.1 Закрепить навык создания запросов в СУБД MySQL.
- 1.2 Научиться создавать SQL-запросы, содержащие встроенные функции MySQL в СУБД MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.677-705.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.736-746.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Вывести на экран текущую базу данных и имя пользователя, используя функции DATABASE() и USER(), текущую дату, используя функции CURDATE(), CURTIME(), NOW().

5.2 Вывести на экран номера заказов текущего года и в отдельных столбцах дату, день и название месяца, когда они были сделаны.

5.3 Заменить все двойные пробелы в названиях книг на одинарные, а цену книг округлить до десятков (например, 123 → 120, 459 → 460).

5.4 Вывести на экран фамилию и первую букву имени автора в первом столбце, название и цену книги с указанием денежной единицы во втором столбце. При выполнении использовать CONCAT или CONCAT_WS.

5.5 Получить выборку данных, содержащую информацию об авторах, у каждого в отдельном столбце должны быть перечислены названия книг без повторов, отсортированные по возрастанию. При выполнении использовать GROUP_CONCAT.

5.6 Вывести на экран информацию о заказчике. Если телефон не указан, вместо него вывести «-». Если вместо адреса прочерк, то в значении должно быть NULL.

5.7 Используя оконную функцию ROW_NUMBER(), вывести все строки таблицы Книги, пронумерованные по порядку. Данные должны быть отсортированы по названию книги.

5.8 Используя оконную функцию DENSE_RANK(), вывести для каждого жанра книги, у которых цена входит в топ-3 самых высоких цен для этого жанра.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 В чем преимущество применения встроенных функций?
- 8.2 Какие виды системных функций имеются в языке MySQL?
- 8.3 Каким образом указываются передаваемые в функцию параметры?
- 8.4 Как вызвать встроенную функцию?

Практическая работа №7

Осуществление полнотекстового поиска в БД MySQL

1 Цель работы

- 1.1 Научиться осуществлять полнотекстовый поиск по таблице, используя SQL-запросы на выборку в MySQL;
- 1.2 Научиться создавать полнотекстовые индексы;
- 1.3 Закрепить навык создания запросов на выборку.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.673-676.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.733-735.
- 2.3 http://www.mysql.ru/docs/man/Fulltext_Search.html
- 2.4 <https://dev.mysql.com/doc/refman/8.0/en/fulltext-search.html>

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Изменить настройки полнотекстового поиска

Для проверки, какое минимально количество символов должно быть в искомом слове, использовать следующую команду:

```
SHOW VARIABLES LIKE 'ft_min_word_len'
```

Для изменения требуется внести изменения в файл конфигурации MySQL:

`C:\...\OpenServer\userdata\config\MySQL-8.0_my.ini`

Указать у `ft_min_word_len` вместо значения 3 значение 2 и перезапустить сервер.

- 5.2 Создать таблицы с полнотекстовыми индексами:

- 5.2.1 Создание таблицы `games`

- создать и заполнить таблицу «`games`», используя скрипт `Games.sql`.
- внести изменения в текст описания так, чтобы в пяти разных играх в описании встречались от 1 до 3 одинаковых слов (например, в первой игре определенное слово встречается 1 раз, во второй то же слово 3 раза, в третьей – 2 раза).
- добавить в таблицу полнотекстовый индекс на столбец Описание.
- добавить в таблицу полнотекстовый индекс на набор столбцов Название и Описание.

5.2.2 Создание таблицы video_games

- создать и заполнить таблицу «video_games», используя скрипт VideoGames.sql.
- добавить в таблицу полнотекстовый индекс на столбец Жанры.

5.3 Определение релевантности

Используя команду MATCH() в командах на выборку всех данных и значения релевантности, осуществить вывод релевантности столбца Описание таблицы «games». Требуется указать в разных командах на выборку

- слово, не встречающееся ни в одном описании,
- слово, встречающееся только в одном описании,
- слово, повторяющееся в разных описаниях,
- два слова.

Образец:

```
SELECT *,  
    MATCH (столбец) AGAINST ('слово1 слово2') AS релевантность  
FROM таблица;
```

5.4 Сортировка по релевантности

Используя команду MATCH() в команде на выборку данных, осуществить поиск строк, содержащий указанный текст в столбце название и/или описание таблицы «games» и вывести значения релевантности, отсортировав по ее убыванию.

Требуется указать в разных командах на выборку

- слово, не встречающееся ни в одном описании/названии,
- слово, встречающееся только в одном описании/названии,
- слово, повторяющееся в разных описаниях/названиях,
- два слова (например, только из описания, только из названия, из описания и названия).

Образец:

```
SELECT *,  
    MATCH (столбец1, столбец2) AGAINST ('слово1 слово2') AS  
релевантность  
FROM таблица  
ORDER BY MATCH (столбец1, столбец2) AGAINST ('слово1 слово2') DESC;
```

5.5 Поиск (фильтрация) совпадений

Используя команду MATCH() в условии команды на выборку данных, осуществить поиск совпадений по двум жанрам таблицы «video_games», отсортировав по убыванию релевантности.

Образец:

```
SELECT *,  
    MATCH (столбец) AGAINST ('слово1 слово2') AS релевантность  
FROM таблица  
WHERE MATCH (столбец) AGAINST ('слово1 слово2');
```

5.6 Используя команду MATCH() в условии команды на выборку данных, осуществить поиск совпадений по столбцу Жанры таблицы «video_games», используя логический режим, позволяющий исключить лишнее слово.

Образец описания списка слов:

AGAINST ('желательное –исключаемое' IN BOOLEAN MODE)

Отобразить значение релевантности для найденных строк.

5.7 Используя команду MATCH() в условии команды на выборку данных, осуществить поиск совпадений по столбцу Жанры таблицы «video_games», используя логический режим, позволяющий сделать обязательным требуемое слово.

Образец описания списка слов:

AGAINST ('желательное +обязательное' IN BOOLEAN MODE)

Отобразить значение релевантности для найденных строк.

5.8 Используя команду MATCH() в условии команды на выборку данных, осуществить поиск совпадений по столбцу Жанры таблицы «video_games», используя логический режим, позволяющий исключить лишнее слово и сделать обязательным другое слово.

Образец описания списка слов:

AGAINST ('желательное +обязательное –исключаемое' IN BOOLEAN MODE)

Отобразить значение релевантности для найденных строк.

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench, подключиться к БД Market.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «полнотекстовый поиск»?

8.2 Для чего применяется команда MATCH()?

8.3 Для чего применяется команда AGAINST()?

8.4 Как включить логический режим в процессе полнотекстового поиска?

8.5 Как создать полнотекстовый индекс?

8.6 В каких случаях требуется создавать полнотекстовый индекс?

8.7 Что показывает значение релевантности в процессе полнотекстового поиска?

Практическая работа №8

Применение транзакций в MySQL

1 Цель работы

- 1.1 Научиться проводить откат и фиксацию транзакций в MySQL;
- 1.2 Закрепить навык создания запросов на выборку и модификацию данных в MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.715-722.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.247-250.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Запустить командную строку, перейти в ней в папку с утилитой mysql:

```
cd c:\...\OpenServer\modules\database\MySQL-8.0\bin
```

Запустить утилиту mysql, введя следующую команду:

```
mysql -h localhost -u root -p
```

Ввести пароль пользователя root.

В случае корректного ввода данных будет получен доступ к командной строке (отобразится приветствие и mysql>)

В командной строке mysql можно писать любые SQL команды к СУБД MySQL. Признаком окончания команды считается ; (точка с запятой).

```
mysql> USE market;
```

```
mysql> SELECT *
```

```
mysql> FROM books;
```

Для поддержки кириллицы в командной строке вызвать команду:

```
set names 'cp866';
```

5.1 В командной строке mysql отключить автовыполнение команд, написав SET AUTOCOMMIT=0 и создать транзакцию, используя команду BEGIN.

5.2 Выполнить команду на добавление записи в таблицу «Авторы» и вывести ее содержимое на экран. Выполнить откат транзакции, используя команду ROLLBACK, и вывести на экран содержимое таблицы «Авторы».

5.3 Выполнить команду на добавление записи в таблицу «Авторы» и вывести ее содержимое на экран. Создать точку сохранения, используя команду SAVEPOINT, и вывести на экран содержимое таблицы «Авторы».

5.4 Выполнить команду на изменение записей таблицы «Авторы» и вывести ее содержимое на экран. Откатить транзакцию к точке сохранения, используя команду ROLLBACK TO SAVEPOINT, и вывести на экран содержимое таблицы «Авторы».

5.5 Выполнить команду на изменение записей таблицы «Авторы» и вывести ее содержимое на экран. Зафиксировать транзакцию, используя команду COMMIT, и вывести на экран содержимое таблицы «Авторы».

6 Порядок выполнения работы

6.1 Подключиться к БД Market, используя командную строку.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «транзакция»?

8.2 Как произвести откат транзакции?

8.3 Как произвести фиксацию транзакции?

8.4 Как создать точку сохранения в рамках транзакции?

Практическая работа №9

Применение планировщика событий

1 Цель работы

1.1 Научиться создавать и настраивать события, автоматически выполняемые на сервере MySQL.

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.673-676.

2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.733-735.

2.3 <https://dev.mysql.com/doc/refman/8.0/en/event-scheduler.html>

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

Создание событий:

- в phpMyAdmin: выбрать БД — пункт верхнего меню События.
- в HeidiSQL: выбрать БД — пункт контекстного меню Создать – Событие.
- в MySQL Workbench: выбрать БД — Schema Inspector – Events.

Все создаваемые события снабжать комментариями, код созданных событий сохранять в файл

5.1 Включение планировщика событий и создание события

5.1.1 До запуска сервера внести следующую настройку в конфигурационный файл MySQL после строки [mysqld]:

```
event_scheduler = on
```

Этот параметр используется для указания статуса (включен или выключен) планировщика событий mysqld.exe (по умолчанию планировщик выключен).

Если сервер был включен, его потребуется перезапустить после модификации конфигурационного файла MySQL.

5.1.2 Для проверки, что планировщик событий (event_scheduler) запущен, выполнить следующую команду:

```
show processlist
```

Если планировщик был включен, то в списке процессов появится event_scheduler.

5.1.3 Создать в БД таблицу myEventTable со следующими столбцами:

- id (целочисленный, автоинкрементный, первичный ключ);

- eventTime (дата и время);
- eventName (символьный, длина 50).

5.1.4 Создать событие «event1», которое будет выполняться каждые 10 секунд на протяжении пяти минут. При наступлении события записывать в предварительно созданную таблицу myEventTable текущие дату и время и название события. По завершении событие не должно удаляться из БД.

5.2 Создать событие «event2», которое будет выполняться каждые две минуты 30 секунд на протяжении суток. При наступлении события записывать в предварительно созданную таблицу myEventTable текущие дату и время и название события. По завершении событие не должно удаляться из БД.

5.3 Создать событие «event3», которое будет выполнено однократно в указанное время. При наступлении события записывать в предварительно созданную таблицу myEventTable текущие дату и время и название события. По завершении событие должно удаляться из БД.

5.4 Создать событие «eventAuthor», которое будет выполняться ежедневно в 15:00. При наступлении события записывать в предварительно созданную таблицу myEventTable текущие дату и время и название события и удалять из таблицы Авторы тех, кто не написал ни одной книги.

5.5 Просмотреть информацию о созданных событиях, используя БД information_schema:

```
SELECT *  
FROM information_schema.EVENTS;
```

Объяснить полученные результаты.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое «событие»?
- 8.2 Каков синтаксис команды создания события?
- 8.3 Какие запросы позволяют просмотреть список событий?
- 8.4 Как задать периодичность и количество выполнений события?

Практическая работа №10

Создание SQL-запросов на модификацию схемы БД в MySQL

1 Цель работы

1.1 Научиться создавать SQL-запросы на создание, изменение и удаление таблиц БД средствами MySQL.

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.624-626.

2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.201-218.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 Написать SQL-запрос, который создает новую таблицу BooksInfo, содержащую данные о книгах (код, название, фамилия автора, имя автора, год выпуска, цена, количество страниц), используя команду CREATE TABLE. Столбцы должны совпадать по типам данных со столбцами таблиц Авторы и Книги. Назначить первичным ключом код книги.

5.2 Написать SQL-запрос, который делает в таблице BooksInfo код книги автоинкрементным.

5.3 Написать SQL-запрос, который добавляет в таблицу BooksInfo ограничение на уникальность названия, имени и фамилии автора (3 столбца одновременно).

5.4 Написать SQL-запрос, который добавляет в таблицу BooksInfo необязательное поле «Дата поступления» для хранения информации о дате поступления книги в магазин.

5.5 Написать SQL-запрос, который удаляет из таблицы BooksInfo поле «Количество страниц».

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench, подключиться к БД Market.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Какое предложение инструкции CREATE TABLE позволяет определить первичный ключ?

8.2 Какое предложение инструкции CREATE TABLE позволяет определить внешний ключ?

8.3 Как в инструкции CREATE TABLE задать ограничения для значений столбца?

8.4 В каких случаях нельзя удалить столбец, таблицу?

8.5 Как сделать значение пары столбцов совместно уникальным?

8.6 Для чего предназначена команда DROP TABLE?

8.7 Что такое «индекс»?

8.8 Каково назначение индексов в БД?

8.9 В каких объектах БД могут быть созданы индексы?

Практическая работа №11

Проведение обратного инжиниринга модели данных из БД MySQL

1 Цель работы

1.1 Научиться проводить обратный инжиниринг БД MySQL, используя CASE-средство MySQL Workbench.

2 Литература

2.1 Мартишин, С.А. Проектирование и реализация баз данных в СУБД MySQL с использованием MySQL Workbench: учебное пособие / С.А. Мартишин, В.Л. Симонов, М.В. Храпченко. – 4-е изд., перераб. и доп. – Москва: ИД| ФОРУМ: НИЦ Инфра-М, 2012. – с.123-130.

3 Подготовка к работе

3.1 Повторить теоретический материал (см. п.2).

3.2 Изучить описание практической работы.

4 Основное оборудование

4.1 Персональный компьютер.

5 Задание

5.1 В MySQL Workbench выбрать пункт меню Database – Reverse Engineer, указать настройки подключения к БД и саму БД. На этапе Select Schemas отметить галочкой БД, в которую надо будет вносить изменения. На этапе Select Objects выбрать все объекты и нажать Execute.

5.2 Полученную модель экспортировать в формате:

- SQL (File – Export – Forward Engineer SQL CREATE Script, в открывшемся окне нажимать Next до тех пор, пока не появится кнопка Save to Other File),

- PNG (File – Export – Export as PNG).

5.3 Во вкладке MySQL Model выбрать в Physical Schema выбрать БД, нажать на нее дважды, затем нажать на Add Table и добавить таблицу Поставка (код книги, количество, дата поставки), настроить в нижней части экрана названия, типы данных и свойства столбцов.

5.4 В сгенерированную модель, используя вкладку EER Diagram, перенести из панели Catalog Tree таблицу Поставка и визуально добавить связь между таблицами Книги и Поставка (в таблице Поставка код книги должен быть внешним ключом для связи с таблицей Книги по столбцу код книги).

5.5 В MySQL Workbench выбрать пункт меню Database – Synchronize Model, указать настройки подключения к БД и саму БД. На этапе Select Changes to Apply указываются вносимые изменения. Сохранить скрипт из этапа Review DB Changes и выполнить его, нажав Execute.

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench, подключиться к БД Market.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего применяется MySQL Workbench?
- 8.2 Что такое «обратный инжиниринг»?
- 8.3 Как провести обратный инжиниринг, используя MySQL Workbench?

Практическая работа №12

Создание представлений в MySQL

1 Цель работы

- 1.1 Научиться создавать и использовать представления в MySQL;
- 1.2 Закрепить навык создания запросов на выборку в MySQL.

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.631-637.

2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-229.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать и протестировать представление, выводящее заказы текущего года (номер, дату, кодЗаказчика, логин, фамилию и имя заказчика).

5.2 Создать и протестировать представление, выводящее информацию о книге (кодКниги, фамилия и имя автора, название, цена).

5.3 Создать и протестировать представление, возвращающее фамилию и имя автора и список написанных им книг, названия разделять точкой с запятой, без дубликатов названий.

5.4 Создать и протестировать представление, использующее в качестве источника данных представление из п.2. После столбца Название добавить оператор CASE, возвращающий 'Да' или 'Нет' в зависимости от того, содержит ли Название слово 'Сказки'.

5.5 Создать и протестировать представление, использующее в качестве источника данных представление из п.2. Перед столбцом Цена добавить оператор CASE, возвращающий ценовую категорию (до 1000, от 1000 до 5000, от 5000).

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое представления и чем они отличаются от таблиц?
- 8.2 Каким целям служат представления?
- 8.3 Каким образом достигается актуальность содержимого представления?
- 8.4 Возможно ли создание представления, включающего информацию из нескольких таблиц одновременно?
- 8.5 Какие требования предъявляются к обновляемым представлениям?
- 8.6 Как создать обновляемое представление?

Практическая работа №13

Создание функций пользователя в MySQL

1 Цель работы

- 1.1 Научиться создавать и использовать скалярные функции пользователя в MySQL;
- 1.2 Закрепить навык создания запросов на выборку в MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.631-637.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-230.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать и протестировать функцию, возвращающую стоимость заказа по номеру заказа (передается в параметрах).
- 5.2 Создать и протестировать функцию, возвращающую строку, содержащую имя и фамилию заказчика, написанные заглавными буквами, по логину заказчика (передается в параметрах).
- 5.3 Создать и протестировать функцию, возвращающую по идентификатору автора (передается в параметрах) в виде одной строки список названий его книг.
- 5.4 Создать и протестировать функцию, возвращающую количество авторов из указанной в параметрах страны.
- 5.5 Создать и протестировать функцию, возвращающую прибыль за указанный в параметрах год.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое функции пользователя и для чего они применяются?
- 8.2 Чем отличается скалярная функция от табличной?
- 8.3 Как объявить переменную и присвоить ей значение?
- 8.4 Как передать параметры в функцию?

Практическая работа №14

Создание хранимых процедур в MySQL

1 Цель работы

- 1.1 Научиться создавать и использовать хранимые процедуры в MySQL;
- 1.2 Закрепить навык создания запросов на выборку и модификацию данных в MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.626-637.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-230.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать и протестировать хранимую процедуру, которая будет добавлять нового заказчика в таблицу Заказчики. Входные параметры: логин, имя, фамилия, адрес, телефон. После добавления процедура должна выводить на экран идентификатор последнего заказчика.

При выполнении использовать функцию `LAST_INSERT_ID()`, которая возвращает последнее значение автоинкрементного столбца, вставленное в таблицу (здесь — идентификатор вставленной записи).

5.2 Создать и протестировать процедуру, возвращающую те строки таблицы Книги, в названиях которых содержится указанный в параметре процедуры текст.

5.3 Создать и протестировать процедуру, добавляющую нового автора (параметры: имя, фамилия, страна). Процедура должна возвращать через параметр код созданного объекта.

5.4 Создать и протестировать хранимую процедуру, изменяющую стоимость всех книг на указанный в параметрах процент (например, параметр 5 — увеличит стоимость на 5%).

5.5 Создать и протестировать хранимую процедуру, удаляющую из таблицы Авторы тех авторов, которые не написали ни одной книги.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Что такое хранимые процедуры и для чего они применяются?
- 8.2 Чем отличается функция пользователя от хранимой процедуры?
- 8.3 Как задать выходные параметры хранимой процедуры?
- 8.4 Кто может вызывать на выполнение хранимую процедуру?

Практическая работа №15

Создание AFTER-триггеров в MySQL

1 Цель работы

- 1.1 Научиться создавать и использовать триггеры в MySQL;
- 1.2 Закрепить навык создания запросов на выборку и модификацию данных в MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.626-637.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-230.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

5.1 Создать таблицу УдаленныеЗаказчики, в который добавить те же столбцы, что и в таблице Заказчики, и столбец для хранения даты удаления записи. Создать AFTER триггер DELETE для таблицы Заказчики, записывающий в таблицу УдаленныеЗаказчики данные из удаленной строки.

5.2 Создать таблицу Logs из полей: таблица, операции, дата и время, текущий пользователь. Создать AFTER триггеры INSERT, UPDATE и DELETE для таблиц Книги и Заказы. При срабатывании триггеров информация о срабатывании должна заноситься в таблицу Logs.

5.3 Изменить AFTER триггер DELETE для таблицы Заказы так, чтобы при его срабатывании удалялись записи о тех заказчиках, которые не сделали ни одного заказа.

5.4 Изменить AFTER триггер INSERT для таблицы Состав так, чтобы при его срабатывании в глобальную переменную @orderCost записывалась стоимость измененного заказа.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое триггер?

8.2 Что хранится в таблицах new и old?

8.3 Когда срабатывает AFTER-триггер?

8.4 Каково назначение AFTER -триггеров?

Практическая работа №16

Создание BEFORE-триггеров в MySQL

1 Цель работы

- 1.1 Научиться создавать и использовать триггеры в MySQL;
- 1.2 Закрепить навык создания запросов на выборку и модификацию данных в MySQL.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.626-637.
- 2.2 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.218-230.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создать BEFORE триггер DELETE для таблицы Заказчики, при удалении заказчика удаляющий предварительно все сделанные им заказы, включая состав заказов.
- 5.2 Создать BEFORE триггер INSERT для таблицы Книги. При срабатывании триггера требуется проверять цену книги, если она больше 5000, то подменять значение цены на 5000.
- 5.3 Создать BEFORE триггер INSERT для таблицы Заказы. При срабатывании триггера вместо указанной даты заказа должна ставиться текущая дата.
- 5.4 Добавить в таблицу Книги столбец Количество (по умолчанию — 100, текущие записи заполнить значением 50). Создать BEFORE триггер INSERT для таблицы Состав, который при заказе книги будет уменьшать количество экземпляров заказанной книги в таблице Книги.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench, подключиться к БД Market.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

8.1 Что такое триггер?

8.2 Чем отличаются триггеры BEFORE и AFTER?

8.3 Когда срабатывает BEFORE-триггер?

8.4 Каково назначение BEFORE-триггеров?

Практическая работа №17

Разграничение прав доступа пользователя к БД MySQL

1 Цель работы

- 1.1 Научиться использовать SQL-команды для управления именами входа и пользователями БД MySQL;
- 1.2 Научиться назначать привилегии пользователю БД;
- 1.3 Закрепить навык создания объектов БД.

2 Литература

2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.622-624.

2.2 <https://dev.mysql.com/doc/refman/8.0/en/grant.html>

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Команды создания пользователя, подключенного с localhost:

CREATE USER 'логин'@'localhost'; # без пароля

CREATE USER 'логин'@'localhost' IDENTIFIED BY 'пароль'; # с паролем

Команды задания привилегий пользователям, подключенным с localhost:

Общая форма:

GRANT привилегия1, привилегия2, ...

ON БД.объектБД

TO 'логин'@'localhost';

Глобальные привилегии (все БД, все объекты БД):

GRANT привилегия1, привилегия2, ...

ON *.*

TO 'логин'@'localhost';

Привилегии уровня БД (на все объекты БД):

GRANT привилегия1, привилегия2, ...

ON БД.*

TO 'логин'@'localhost';

Привилегии уровня таблицы БД:

GRANT привилегия1(поле1, поле2, ...), привилегия2(поле1, поле2, ...), ...

ON БД.таблица

TO 'логин'@'localhost';

5.1 Создать пользователя **userTask1** у которого есть разрешение уровня сервера на просмотр списка БД (привилегия SHOW DATABASES).

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.2 Создать пользователя **userTask2**, у которого есть все привилегии уровня сервера (привилегия ALL PRIVILEGES). Указать при создании пользователя пароль 123.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.3 Создать пользователя **userTask3** с паролем qwerty, имеющего привилегии на выполнение DML команд (SELECT, INSERT, UPDATE, DELETE) в БД Market.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.4 Создать пользователя **userTask4**, имеющего привилегии на выполнение запросов на выборку в таблице Books БД Market.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.5 Создать пользователя **userTask5**, имеющего привилегии на выполнение запросов на выборку из столбцов код, название и цена и на изменение столбца цена таблицы Books БД Market.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.6 Открыть phpmyadmin. Нажать на вкладку «Учетные записи пользователей». Ознакомиться со списком доступных привилегий у имеющихся пользователей.

При нажатии на кнопку «Экспорт» в таблице пользователей отображаются команды для задания привилегий.

При нажатии на кнопку «Редактировать привилегии» в таблице пользователей отображается окно для задания привилегий (во вкладке Глобальные – привилегии уровня сервера, во вкладке База данных – привилегии уровня БД).

Нажать кнопку «Добавить пользователя», создать пользователя **userTask6** без пароля. Разрешить ему следующие глобальные привилегии: SELECT, SHOW DATABASES

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.7 Используя phpMyAdmin, создать пользователя **userTask7**. Разрешить ему следующие привилегии в БД Market: просмотр таблиц Книги и Авторы, вставку записей в таблицу Книги. Для разрешения указывается БД и в ней отмечаются требуемые команды.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

5.8 Используя MySQL Workbench, создать пользователя **userTask8** с паролем 12345. Назначить ему права на выборку данных из БД Market. Для перехода к созданию пользователя используется вкладка Administration, раздел Users and Privileges, в котором требуется нажать Add Account и указать настройки пользователя.

Подключиться под созданным пользователем, используя MySQL Workbench, и проверить, какие действия он может выполнить.

6 Порядок выполнения работы

- 6.1 Запустить MySQL Workbench.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего требуется создавать пользователей БД?
- 8.2 Какие виды угроз БД существуют?
- 8.3 Какой вид угроз БД позволяет устранить разграничение доступа?
- 8.4 Как идентифицируются пользователи в MySQL?
- 8.5 На какие уровни разделяется система безопасности MySQL?
- 8.6 Какие виды привилегий можно назначить пользователю БД?

Практическая работа №18

Резервное копирование и восстановление БД MySQL

1 Цель работы

- 1.1 Научиться обеспечивать безопасность БД путем применения резервного копирования и восстановления;
- 1.2 Закрепить навык работы с объектами БД в MySQL.

2 Литература

- 2.1 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.252-256.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

Создание резервной копии:

- в phpMyAdmin: выбрать БД — пункт Экспорт верхнего меню.
- в HeidiSQL: меню Инструменты— Экспорт базы данных в SQL.
- в MySQL Workbench: Administration — Data Export.

Восстановление из резервной копии:

- в phpMyAdmin: выбрать БД — пункт Импорт верхнего меню.
- в HeidiSQL: меню Инструменты — Импорт CSV файла.
- в MySQL Workbench: Administration — Data Import/Restore.

5.1 Создать резервную копию таблиц БД, используя экспорт данных таблицы в формате csv

5.2 Изменить записи в таблице БД, для которой был выполнен экспорт, и восстановить исходные значения, выполнив импорт из файла в таблицу БД

5.3 Создать резервную копию таблиц БД, используя экспорт всех таблиц и триггеров в формате sql (только CREATE)

5.4 Создать резервную копию таблиц БД, используя экспорт всех таблиц и их данных в формате sql (CREATE и INSERT)

5.5 Создать резервную копию всех объектов БД, используя экспорт в формате sql (только CREATE)

5.6 Создать новую БД и создать в ней таблицы и триггеры, используя скрипт из п.5.3

5.7 Создать новую БД и создать в ней заполненные таблицы, используя скрипт из п.5.4

5.8 Создать новую БД и создать в ней объекты БД, используя скрипт из п.5.5

6 Порядок выполнения работы

- 6.1 Запустить OpenServer, подключиться к БД Market.
- 6.2 Выполнить задания из п.5, используя различные средства для экспорта и импорта данных (MySQL Workbench, phpMyAdmin, HeidiSQL).
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Для чего создаются резервные копии БД?
- 8.2 В чем отличие между полным и потабличным резервным копированием?
- 8.3 Где может быть сохранена резервная копия?
- 8.4 Как часто должно выполняться резервное копирование БД?
- 8.5 Какие способы создания резервных копий применяются в MySQL?
- 8.6 Как восстановить БД MySQL из резервной копии?

Практическая работа №19

Разработка приложения для авторизации пользователей MySQL

1 Цель работы

- 1.1 Научиться разрабатывать приложения для авторизации и регистрации пользователей MySQL;
- 1.2 Научиться создавать приложения для доступа к данным.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.650-661.
- 2.2 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. – 2-е изд. – Санкт-Петербург: Питер, 2014. – с.191-240.
- 2.3 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.258-287.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание на C# слоя доступа к данным и создание строки подключения
 - 5.1.1 Создать оконное приложение на C#. Добавить в него ссылку на библиотеку для работы с MySQL:
Проект – Ссылки – PKM – Сборки – найти и добавить библиотеку MySql.Data
 - 5.1.2 Создать статический класс DAL (DataAccessLayer), в котором будет храниться логика доступа к данным из клиентского приложения. В разделе using подключить следующие пространства имен:
MySql.Data.MySqlClient // для работы с MySQL из приложения C#
System.Data // для работы с DataTable и DataSet
 - 5.1.3 Добавить в класс DAL открытые строковые поля для хранения названия сервера, порта, БД, логина и пароля пользователя, присвоить им значения по умолчанию (указать свои настройки для подключения к БД).
 - 5.1.4 Добавить в класс DAL открытый метод (можно заменить на свойство только на чтение), возвращающий строку подключения к БД MySQL. Для реализации использовать значения полей класса и объект MySqlConnectionStringBuilder:
var stringBuilder = new MySqlConnectionStringBuilder();
stringBuilder.Server = хост;
stringBuilder.UserID = пользователь;
stringBuilder.Password = пароль;
stringBuilder.Database = БД;
return stringBuilder.ConnectionString;

5.1.4 Проверить работу созданного метода, вызвав его в оконном приложении, полученную строку отобразить на экране.

5.2 Добавить в приложение на C# форму авторизации. На форме должны быть поля для ввода логина и пароля и кнопка «Войти».

При вводе логина и пароля после нажатия кнопки «Войти» должны изменяться значения логина и пароля в настройках подключения (поля класса DAL) и выводиться сообщение «Здравствуйте, логин!», если к БД можно подключиться с этими данными.

Если пользователя с указанными логином и паролем в БД отсутствуют (т.е. при попытке подключения выдается ошибка), то сценарий должен выводить соответствующее сообщение.

5.3 Создать приложение на Python:

5.3.1 Импортировать модуль Python DB-API в приложение и настроить подключение к БД MySQL (название модуля данных: mysql.connector):

```
# Импортируем библиотеку, соответствующую типу базы данных
import модульДанных
# Создаем соединение с БД
conn = модульДанных.connect(host="localhost", #Адрес mysql сервера
    user="пользователь mysql сервер",
    passwd="Пароль пользователя",
    db="БД")
```

5.3.2 Создать в приложении курсор (объект, предназначенный для выполнения запросов и получения их результатов):

```
# Открываем соединение с БД
cursor = conn.cursor()
# после создания курсора и до закрытия подключения выполнять работу с БД
```

5.3.3 Закрывать соединение с БД

```
# Закрываем соединение с БД
conn.close()
```

5.3.4 Реализовать перехват исключений

6 Порядок выполнения работы

6.1 Запустить MySQL Workbench и MS Visual Studio.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.3 Ответы на контрольные вопросы

7.4 Вывод

8 Контрольные вопросы

8.1 Что такое «авторизация»?

8.2 Что такое «регистрация»?

8.3 В каких случаях рекомендуется создавать таблицу БД, хранящую информацию о зарегистрированных пользователях сайта, вместо создания пользователей БД?

Практическая работа №20

Разработка приложения для выполнения запросов на выборку данных из БД MySQL

1 Цель работы

- 1.1 Научиться разрабатывать приложения с графическим интерфейсом пользователя для выполнения запросов на выборку данных из БД MySQL;
- 1.2 Научиться создавать приложения для доступа к данным.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.650-661.
- 2.2 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. – 2-е изд. – Санкт-Петербург: Питер, 2014. – с.242-274.
- 2.3 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.288-309.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание на C# метода для выборки набора значений

5.1.1 Добавить в класс DAL открытый метод, принимающий строку с текстом SQL-команды, выполняющий ее и возвращающий полученный результат как DataTable:

```
DataTable table = new DataTable();  
using (MySqlConnection connection = new MySqlConnection(подключение))  
{  
    connection.Open();  
    MySqlCommand command = new MySqlCommand(запрос, connection);  
    table.Load(command.ExecuteReader());  
}  
return table;
```

5.1.2 Проверить работу созданного метода, вызвав его в оконном приложении. Текст SQL-команды должен вводиться пользователем в поле ввода, результат выполнения команды должен отображаться на экране. Результат отобразить в DataGridView:

контрол.DataSource = данные;

5.1.3 Добавить на форму кнопку «Показать таблицу», поле ввода и DataGridView. Реализовать вывод содержимого таблицы БД в компоненте

DataGridView после нажатия на кнопку «Показать таблицу». Название таблицы должен указывать пользователь в поле ввода.

5.2 Создание на C# метода для выборки одного значения

5.2.1 Добавить в класс DAL открытый метод, принимающий строку с текстом SQL-команды, выполняющий ее и возвращающий полученное значение как объект.

Для реализации использовать MySqlCommand.ExecuteScalar().

5.2.2 Проверить работу созданного метода, вызвав его в оконном приложении. Текст SQL-команды должен вводиться пользователем в поле ввода, результат выполнения команды должен отображаться на экране.

5.3 Реализовать на Python чтение данных из указанной пользователем таблицы:

Использовать скрипт из ПР №19. Перед закрытием соединения дописать следующий код:

```
# Выполняем SELECT запрос к базе данных после открытия соединения
cursor.execute("SELECT * FROM НазваниеТаблицы")
# Получаем и выводим на экран результат запроса
results = cursor.fetchall()
print(results)
```

Добавить построчный вывод только названия и цены книги.

5.4 Реализовать на Python поиск записей из таблицы Книги с ценой, меньше указанной пользователем, используя в запросе подстановки значений по порядку на места подстановок (обозначены подстрокой %s):

```
cursor.execute("SQL команда, содержащая %s", (кортеж значений подстановки))
```

6 Порядок выполнения работы

6.1 Запустить MS Visual Studio.

6.2 Выполнить задания из п.5.

6.3 Ответить на контрольные вопросы.

7 Содержание отчета

7.1 Титульный лист

7.2 Цель работы

7.2 Ответы на контрольные вопросы

7.3 Вывод

8 Контрольные вопросы

8.1 Каково назначение элемента MySqlConnection?

8.2 Каково назначение элемента MySqlCommand?

8.3 Каково назначение элемента MySqlDataReader?

8.4 Какое пространство имен требуется подключить для подключения к СУБД MySQL Server?

Практическая работа №21

Разработка приложения для выполнения запросов на модификацию данных в БД MySQL

1 Цель работы

- 1.1 Научиться разрабатывать приложения с графическим интерфейсом пользователя для выполнения запросов на модификацию данных в БД MySQL;
- 1.2 Научиться создавать приложения для доступа к данным.

2 Литература

- 2.1 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL. Джентельменский набор Web-мастера / Н.А. Прохоренок, В.А. Дронов. – 4-е изд., перераб. и доп. – Санкт-Петербург: БХВ-Петербург, 2015. – с.650-661.
- 2.2 Маклафлин Б. PHP и MySQL. Исчерпывающее руководство. – 2-е изд. – Санкт-Петербург: Питер, 2014. – с.242-274.
- 2.3 Никсон Р. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. – 4-е изд. – Санкт-Петербург: Питер, 2016. – с.288-309.

3 Подготовка к работе

- 3.1 Повторить теоретический материал (см. п.2).
- 3.2 Изучить описание практической работы.

4 Основное оборудование

- 4.1 Персональный компьютер.

5 Задание

- 5.1 Создание на C# метода для выполнения SQL-команд

5.1.1 Добавить в класс DAL открытый метод, принимающий строку с текстом SQL-команды, выполняющий ее и возвращающий количество измененных командой строк.

Для реализации использовать MySqlCommand.ExecuteNonQuery() – метод выполняет SQL-команду и возвращает количество измененных в команде строк.

5.1.2 Проверить работу созданного метода, вызвав его в оконном приложении. Текст SQL-команды должен вводиться пользователем в поле ввода, результат выполнения команды должен отображаться на экране.

5.2 Создать на C# форму удаления записи из таблицы. На форму добавить элементы управления, позволяющие удалить книгу из таблицы Книги БД Market по коду книги, указанному пользователем.

5.3 Создать на C# форму добавления новой записи в таблицу. На форму добавить элементы управления, позволяющие добавить новую строку в таблицу Книги БД Market, значения столбцов должны указываться пользователем.

5.4 Выполнить на Python команду на модификацию данных БД, изменив цену книги с определенным кодом (код и цена указываются пользователем):

```
# Выполняем DML-запрос к БД
cursor.execute("DML-запрос")
# Сохраняем транзакцию для изменения данных в БД
conn.commit()
# Проверяем результат
cursor.execute("запрос на выборку данных")
results = cursor.fetchall()
print(results)
```

6 Порядок выполнения работы

- 6.1 Запустить MS Visual Studio.
- 6.2 Выполнить задания из п.5.
- 6.3 Ответить на контрольные вопросы.

7 Содержание отчета

- 7.1 Титульный лист
- 7.2 Цель работы
- 7.3 Ответы на контрольные вопросы
- 7.4 Вывод

8 Контрольные вопросы

- 8.1 Какие методы MySqlCommand применяются для выполнения SQL-команд и что эти методы возвращают?
- 8.2 Как выполнить набор команд по работе с БД MySQL в одной транзакции?
- 8.3 Какая функция в Python применяется для выполнения запросов на модификацию данных в БД?