МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ   
РОССИЙСКОЙ ФЕДЕРАЦИИ  
  
Федеральное государственное бюджетное образовательное учреждение   
высшего профессионального образования   
«Самарский национальный исследовательский университет   
имени академика С.П. Королёва»

(Самарский университет)  
  
Факультет информатики  
Кафедра программных систем  
  
Дисциплина  
**Практикум на ЭВМ  
  
  
  
ОТЧЕТ**

Проектирование базы данных.  
  
Вариант №26

Студенты: Лапко Я.А. Поборознюк П.Д.   
Группа: 6313 – 020302D  
  
Преподаватель: Попова-Коварцева Д. А.  
  
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Самара 2019

**Постановка задачи:**

В БД должна храниться информация:

* о ПИСАТЕЛЯХ: номер паспорта, фамилия, имя, отчество, домашний адрес, телефон;
* КОНТРАКТАХ: номер контракта, дата заключения контракта, срок контракта (лет), контракт расторгнут или не расторгнут (Да/Нет), дата расторжения контракта
* КНИГАХ: шифр книги, название, тираж, дата выхода из печати, себестоимость (руб.), цена продажи (руб.), гонорар (на всех авторов книги, тыс. руб.);
* ЗАКАЗЧИКАХ: название заказчика, адрес, телефон, обращаться к (Ф.И.О.);
* ЗАКАЗАХ: номер заказа, дата поступления заказа, дата выполнения заказа, количество экземпляров заказываемой книги.

При проектировании БД необходимо учитывать следующее:

* писателю соответствует один контракт (в случае перезаключения контракта прежний контракт заменяется новым). Контракт заключается с одним писателем;
* писатель может написать несколько книг. Книга может быть написана несколькими писателями;
* на книгу может быть сделано несколько заказов. Заказ оформляется на одну

книгу;

* заказчик может сделать несколько заказов. Заказ оформляется на одного заказчика.

Кроме того следует учесть:

* каждому писателю обязательно соответствует контракт. Каждый контракт

обязательно заключается с писателем;

* писатель не обязательно пишет книгу (он заключил контракт, но по какой-то причине не пишет книгу). Каждая книга обязательно пишется одним или несколькими писателями;
* на книгу не обязательно делается заказ (она может быть не востребована заказчиками). Каждый заказ обязательно оформляется на книгу;
* каждый заказчик обязательно делает заказ. Каждый заказ обязательно

оформляется на заказчика

**Тема работы 1:** Проектирование базы данных.

**Содержание работы:**

1. Концептуальное проектирование. Разработать ER-модель предметной области, описанной в проекте. Каждую сущность охарактеризовать набором атрибутов.

2. Логическое проектирование. Преобразовать ER-модель в реляционную модель. С помощью CASE-средства разработать логическую модель БД.

3. Физическое проектирование. Создать в выбранной СУБД разработанную БД.

4. Оформите отчет о выполнении лабораторной работы, включающий описание логической модели данных.

**Ход выполнения работы:**

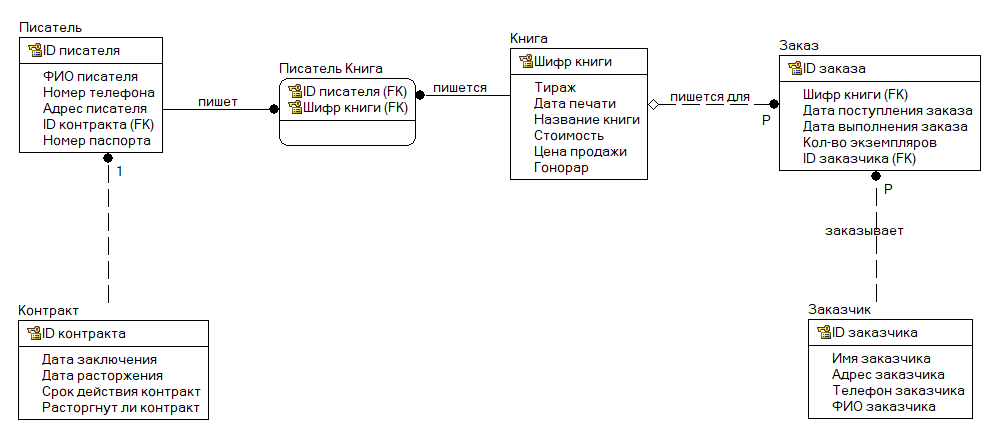
На рисунке 1 представлена логическая модель предметной области.

Рисунок 1 – Логическая модель заданной предметной области

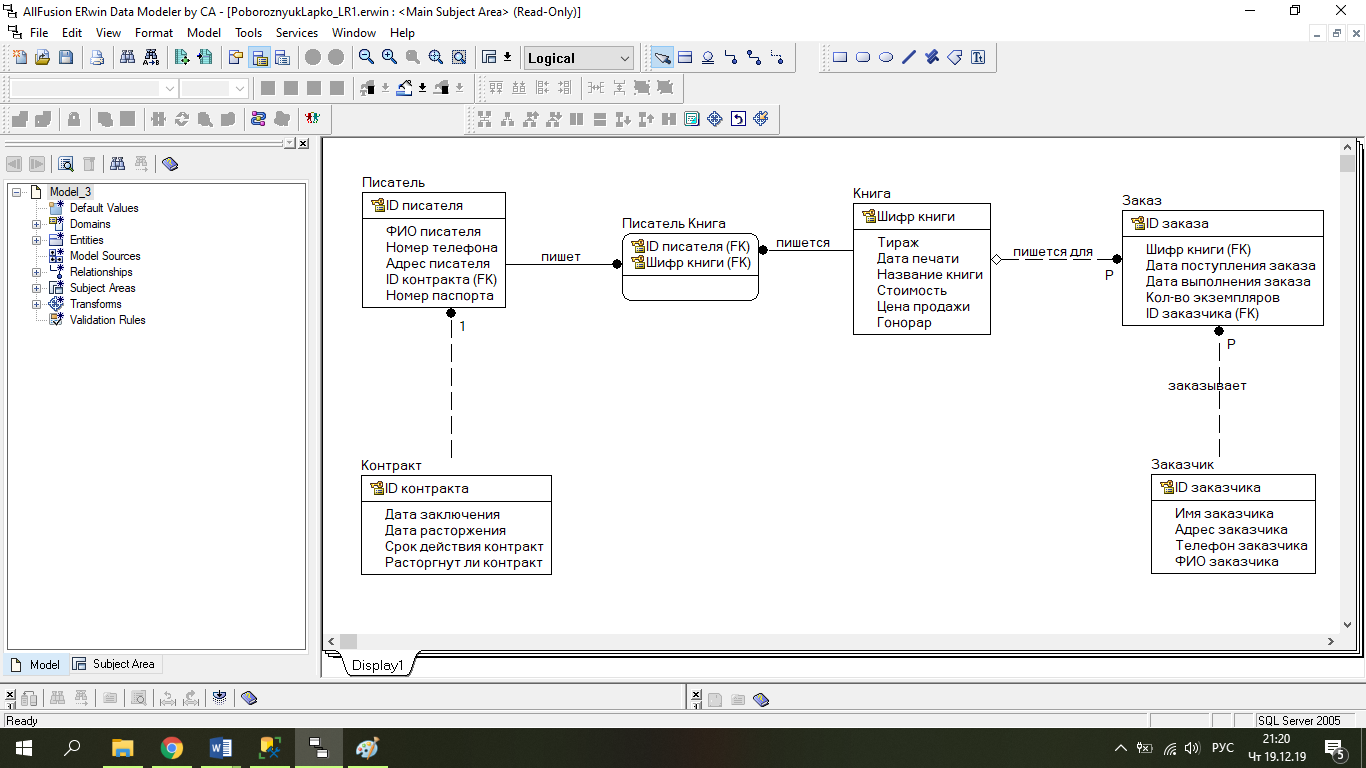


Рисунок 2 – Физическая модель заданной предметной области

SQL скрипт:

CREATE TABLE Book

(

cipher\_book integer NOT NULL ,

circ integer NULL ,

print\_date datetime NULL ,

title\_b varchar(100) NULL ,

cost integer NULL ,

sell\_price integer NULL ,

fee integer NULL

)

go

ALTER TABLE Book

ADD CONSTRAINT XPKКнига PRIMARY KEY CLUSTERED (cipher\_book ASC)

go

CREATE TABLE Contract

(

ID\_contr integer NOT NULL ,

contr\_date datetime NULL ,

contr\_term\_date datetime NULL ,

duration integer NULL ,

is\_term integer NULL

)

go

ALTER TABLE Contract

ADD CONSTRAINT XPKКонтракт PRIMARY KEY CLUSTERED (ID\_contr ASC)

go

CREATE TABLE Customer

(

ID\_cust integer NOT NULL ,

name\_cust varchar(100) NULL ,

address\_cust varchar(100) NULL ,

phone\_num\_cust varchar(20) NULL ,

FN\_cust varchar(100) NULL

)

go

ALTER TABLE Customer

ADD CONSTRAINT XPKЗаказчик PRIMARY KEY CLUSTERED (ID\_cust ASC)

go

CREATE TABLE Ord

(

ID\_order integer NOT NULL ,

d\_receipt\_ord datetime NULL ,

d\_compl\_ord datetime NULL ,

num\_copies integer NULL ,

cipher\_book integer NULL ,

ID\_cust integer NOT NULL

)

go

ALTER TABLE Ord

ADD CONSTRAINT XPKЗаказ PRIMARY KEY CLUSTERED (ID\_order ASC)

go

CREATE TABLE Writer

(

ID\_writer integer NOT NULL ,

FN\_writer varchar(100) NULL ,

ph\_num varchar(20) NULL ,

address\_writ varchar(100) NULL ,

ID\_contr integer NOT NULL ,

passp\_num varchar(20) NULL

)

go

ALTER TABLE Writer

ADD CONSTRAINT XPKПисатель PRIMARY KEY CLUSTERED (ID\_writer ASC)

go

CREATE TABLE Writer\_Book

(

ID\_writer integer NOT NULL ,

cipher\_book integer NOT NULL

)

go

ALTER TABLE Writer\_Book

ADD CONSTRAINT XPKПисатель\_Книга PRIMARY KEY CLUSTERED (ID\_writer ASC,cipher\_book ASC)

go

ALTER TABLE Ord

ADD CONSTRAINT Делает FOREIGN KEY (cipher\_book) REFERENCES Book(cipher\_book)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Ord

ADD CONSTRAINT делается FOREIGN KEY (ID\_cust) REFERENCES Customer(ID\_cust)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Writer

ADD CONSTRAINT Заключён FOREIGN KEY (ID\_contr) REFERENCES Contract(ID\_contr)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Writer\_Book

ADD CONSTRAINT пишет FOREIGN KEY (ID\_writer) REFERENCES Writer(ID\_writer)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

ALTER TABLE Writer\_Book

ADD CONSTRAINT пишется FOREIGN KEY (cipher\_book) REFERENCES Book(cipher\_book)

ON DELETE NO ACTION

ON UPDATE NO ACTION

go

**Тема работы 2:** Конструирование запросов.

**Содержание работы:**

1. Сформулировать и сконструировать в выбранной СУБД *запросы* к БД, которые будут необходимы для предметной области (в соответствии с вариантом задания):
   * Запрос на выборку избранных полей таблицы, с использованием синонима (алиаса) и сортировкой записей (ORDER BY).
   * Запрос с использованием сортировки (ORDER BY) и группировки (GROUP BY).
   * Запрос с использованием предложения DISTINCT.
   * Запрос с использованием операций сравнения.
   * Запросы для предикатов: IN, BETWEEN, LIKE, IS NULL.
   * Запросы с использованием агрегатных функций (COUNT, SUM, AVG, MAX, MIN ), производящие обобщенную групповую обработку значений полей (используя ключевые фразы GROUP BY и HAVING).
   * Запрос на выборку данных из двух связанных таблиц. Выбрать несколько полей, по которым сортируется вывод.
   * Многотабличный запрос с использованием внутреннего и внешнего соединения.
   * Многотабличный запрос с использованием оператора UNION.
2. Создайте SQL команды для модификации данных (INSERT, UPDATE, DELETE).
3. Оформите отчет о выполнении лабораторной работы, включающий описание SQL-запросов и результатов их выполнения.

**Ход выполнения работы**.

SELECT AVG(duration) as average, is\_term

FROM Contract

GROUP BY is\_term

HAVING SUM(duration) > 10



Рисунок 3 – Среднее значение срока контракта для расторгнутых и не расторгнутых контрактов

SELECT cipher\_book, COUNT(\*) AS amount\_of\_autors

FROM Writer\_book

GROUP BY cipher\_book

HAVING COUNT(\*) > 1



Рисунок 4 – Шифры книг, которые писали два автора

Select title\_b from book where title\_b like 'П%'



Рисунок 5 – Все книги, начинающиеся на “П”

Select FN\_writer, cost as book\_cost from

(book FULL OUTER JOIN Writer\_book on book.cipher\_book = Writer\_book.cipher\_book)

FULL OUTER JOIN writer on Writer\_book.ID\_writer = Writer.ID\_writer where cost > 500



Рисунок 6 – Авторы и их книги, у которых стоимость больше 1000

select FN\_writer from writer UNION select FN\_cust from Customer

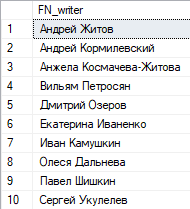


Рисунок 7 – Имена всех авторов и заказчиков

SELECT fee, SUM(sell\_price) as sum\_sell\_price

FROM Book

GROUP BY fee

HAVING SUM(sell\_price) > 200

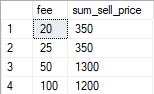


Рисунок 8 – Гонорар и цена продажи каждой книги, что стоимость

больше 200

SELECT \*

FROM Book

ORDER BY cost DESC

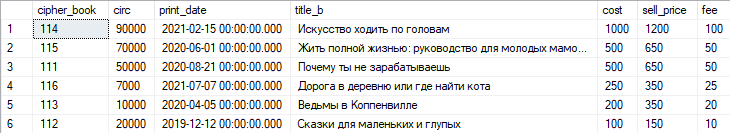


Рисунок 9 – Вывод таблицы “Book” с отсортированной

по убыванию ценой

SELECT distinct contr\_term\_date FROM Contract order by contr\_term\_date

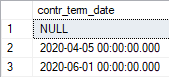


Рисунок 10 – Вывод всех дат расторжения контрактов без повторения

Select ID\_contr from Contract where contr\_term\_date is NULL



Рисунок 11 – ID контрактов, которые не были расторгнуты

insert into Writer\_book values (7, 111)

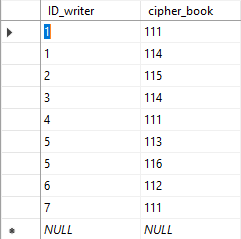


Рисунок 9 – Новая запись

UPDATE Writer\_Book

SET cipher\_book = 112

WHERE ID\_writer = 7

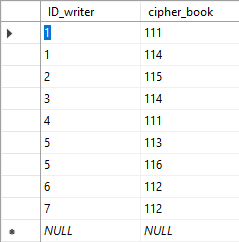
****

Рисунок 12 – Обновление цены записи

Delete from Writer\_Book where ID\_writer = 7

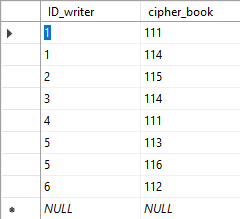


Рисунок 13 – Запись удалена

**Тема работы3:** Разработка интерфейса пользователя для работы с данными в БД. Создание форм.

**Содержание работы:**

Изучите теоретические основы разработки форм, рассмотренные в методических указаниях или других источниках.

1. Создайте необходимые формы для ввода информации в базу данных (созданной на предыдущих лабораторных работах), согласно предметной области своего варианта.
2. Проверьте работу форм (введите, измените и удалите около 10 записей в каждой форме).
3. Проверьте правильность работы обеспечения целостности данных.
4. Обдумайте и создайте формы, которые, возможно, будут полезными для будущих пользователей вашей БД.
5. Оформите отчет о выполнении лабораторной работы (описать каждую разработанную форму, как объекты БД).

**Ход работы:**

Программа состоит из одной формы, способной получать таблицы из базы данных и сохранять изменения, производимые в них обратно в базу данных. Пользователь выбирает название таблицы из выпадающего списка и проводит работу с отношением БД. Пользователь также может выбрать представление из БД для просмотра, например, гонораров писателям, как показано на рисунках ниже.



Рисунок 14 – Начальная форма

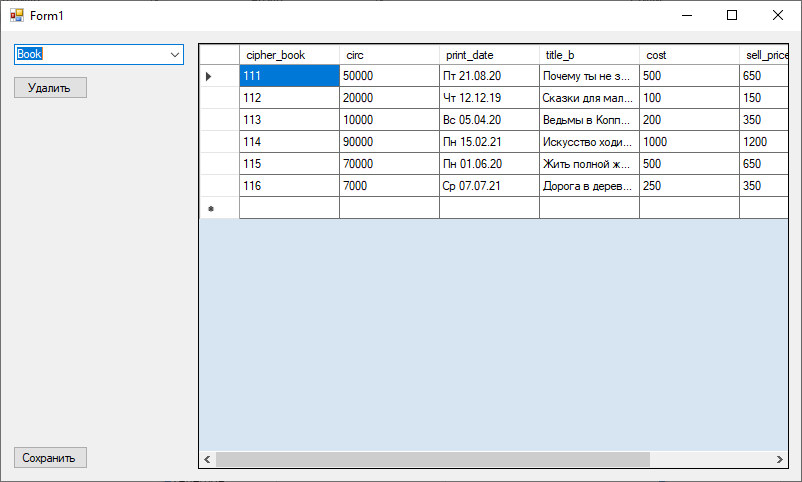


Рисунок 15 – Вывод таблиц

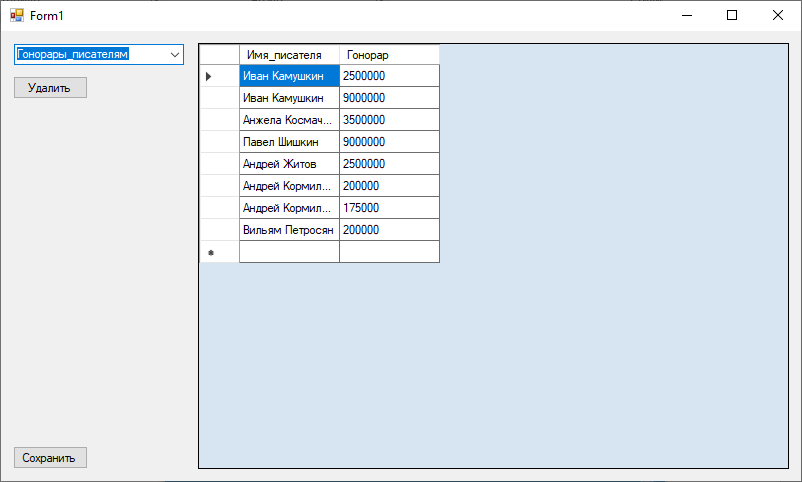


Рисунок 16 – Вывод представления

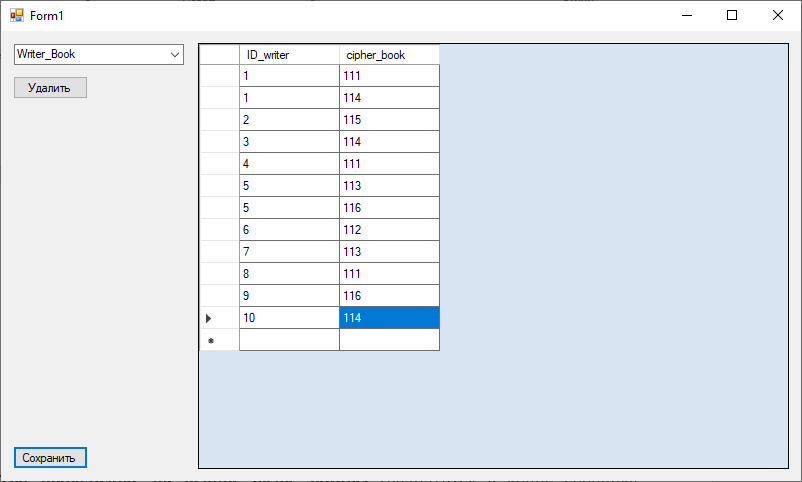


Рисунок 17 – Добавление записей в таблицу и сохранение

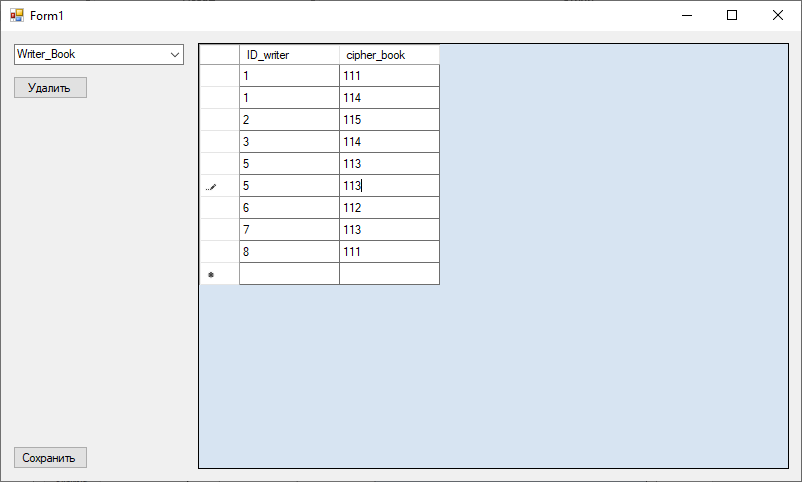


Рисунок 18 – Удаление выбранной строки из таблицы

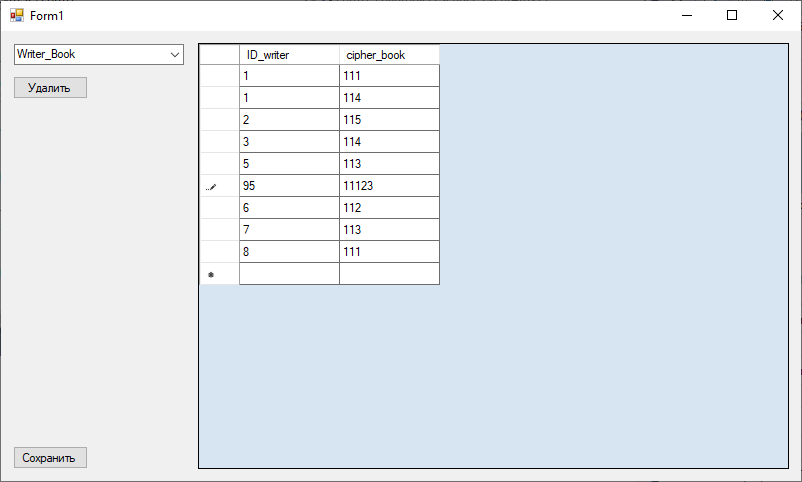


Рисунок 19 – Изменение записей в таблице

**Тема 4:** Конструирование отчета

**Цель работы:** Приобретение навыков по конструированию отчета, основанного на данных, содержащихся в базе данных.

**Содержание работы:**

* 1. Сконструировать отчет, вид которого следует выбрать из раздела "Отчеты, выводимые на основе базы данных" (находится в конце сборника заданий) согласно номеру своего проекта. В отчете произвести вычисления в строках и подвести частные и общие итоги (если они указаны).
  2. Сформировать отчет о выполнении лабораторной работы (описание сконструированного отчета БД), он должен иметь примечание, в котором следует указать данные о разработчике – Ф.И.О. и шифр группы.

## Прибыль от продаж книг издательского центра "Печать"

### за год

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название книги | Себестоимость, руб. | Цена продажи, руб. | Количество экземпляров | Прибыль от продажи книги, руб. |

*Заказчик* ***\_***

|  |  |  |  |
| --- | --- | --- | --- |
| **. .** | **.** | **.** | **?** |
| *Итого получено от заказчика:* |  |  | **?** |

## . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

**Итого: ?**



Примечание. "Прибыль от продажи книги, руб." = ("Цена продажи, руб." – "Себестоимость, руб.") x "Количество экземпляров".

Описание отчёта БД:

Отчёт был составлен на основе sql запроса, приведённого далее. Результат его выполнения показан на рисунке ниже.

Select book.title\_b as Название книги, book.cost as себестоимость, book.sell\_price as Цена продажи, ord.num\_copies as количество экземпляров, (sell\_price-cost)\*num\_copies as прибыль от продажи книги, name\_cust as заказчик from Customer left join Ord on Customer.ID\_cust = ord.ID\_cust left join book on Ord.cipher\_book = book.cipher\_book

**Примечание**

Работу выполнили   
студенты группы №6313-20302D:  
Поборознюк П.Д., Лапко Я.А.