

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Программное обеспечение информационных технологий»

БАЗЫ И БАНКИ ДАННЫХ

*Методические рекомендации к самостоятельной работе
для студентов специальности 1-53 01 02 «Автоматизированные
системы обработки информации»
заочной формы обучения*



Могилев 2022

УДК 004.65
ББК 32.81
Б17

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Программное обеспечение информационных технологий» «08» апреля 2022 г., протокол № 10

Составители: канд. техн. наук, доц. К. В. Захарченков;
канд. техн. наук, доц. Т. В. Мрочек

Рецензент канд. техн. наук, доц. В. М. Ковальчук

Изложены рекомендации по решению задач к аудиторной контрольной работе, приведены примеры решения задач, условия задач для самостоятельного решения в контрольной работе, а также список рекомендуемой литературы. Предназначены для выполнения самостоятельной работы студентами специальности 1-53 01 02 «Автоматизированные системы обработки информации» заочной формы обучения.

Учебно-методическое издание

БАЗЫ И БАНКИ ДАННЫХ

Ответственный за выпуск	В. В. Кутузов
Корректор	А. А. Подошевка
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 21 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

1 Методические указания к выполнению аудиторной контрольной работы.....	4
2 Варианты заданий	11
Список литературы	16

1 Методические указания к выполнению аудиторной контрольной работы

1.1 Создание представлений

Представление – это именованный запрос на выборку, сохраненный в базе данных, который выглядит и работает как таблица, при обращении по имени создает виртуальную таблицу, наполняя ее актуальными данными из базы данных. С представлением не ассоциированы никакие данные, в отличие от таблицы, поэтому представление иногда называют виртуальной таблицей, с которой можно работать так же, как с реально существующей на диске таблицей. Физически представление реализовано в виде SQL-запроса, на основе которого производится выборка данных из одной или нескольких таблиц или представлений.

Представление может выбирать данные из других представлений, которые, в свою очередь, могут также основываться на представлениях или таблицах. Вложенность представлений не должна превышать 32.

При обращении к представлению сервер проверяет правильность всех ссылок в запросе. Проверяется, существуют ли объекты (таблицы и другие представления), требующиеся для выполнения запроса, определяющего представление. Если одна из таблиц, на которые ссылается запрос, была уничтожена, то представление будет нельзя использовать и при попытке обратиться к нему пользователи получат сообщение об ошибке.

Механизм представлений в SQL Server является реализацией одного из 12 принципов Э. Кодда «База данных должна быть доступна конечным пользователям только через представления», что позволяет предоставить пользователю только ту часть БД, которая ему действительно нужна для работы и скрыть концептуальную схему, доступную только администратору базы данных. Поэтому представление часто применяется для ограничения доступа пользователей к конфиденциальным данным в таблице.

Для создания представлений средствами SQL используется следующая конструкция:

```
CREATE VIEW view_name [(column [...n])]
[WITH ENCRYPTION]
AS
select_statement
[WITH CHECK OPTION]
```

Рассмотрим составляющие данной конструкции.

`view_name` – имя представления. При указании имени необходимо придерживаться тех же правил и ограничений, что и при создании таблицы.

`column` – имя колонки, которое будет использоваться в представлении. Максимальная длина имени составляет 128 символов. По умолчанию имена колонок в представлении соответствуют именам колонок в исходных таблицах.

Явное указание имени колонки требуется при использовании вычисляемых колонок или при объединении нескольких таблиц, имеющих колонки с одинаковыми именами. Имена колонок перечисляются через запятую в соответствии с их порядком в представлении. Имена колонок можно указывать в команде SELECT, определяющей представление.

WITH ENCRYPTION – использование этого параметра предписывает серверу шифровать код SQL-запроса. Это гарантирует, что пользователи не смогут просмотреть код запроса и использовать его. Если при определении представления необходимо скрыть имена исходных таблиц и колонок, а также алгоритм объединения данных, то необходимо использовать эту опцию.

select_statement – код запроса SELECT, выполняющий выборку, объединение и фильтрацию строк из исходных таблиц и представлений. Можно использовать команду SELECT любой сложности со следующими ограничениями:

- 1) нельзя создавать новую таблицу на основе результатов, полученных в ходе выполнения запроса, т. е. запрещается использование параметра INTO;
- 2) нельзя проводить выборку данных из временных таблиц, т. е. нельзя использовать имена таблиц, начинающихся на # или ##;
- 3) в представлении нельзя включать предложение ORDER BY, если только в списке выбора инструкции SELECT нет также предложения TOP.

Предложение WITH CHECK OPTION применяется только к обновлениям, выполненным через представление. Оно неприменимо к обновлениям, выполненным непосредственно в базовых таблицах представления. Если имеется представление с ограничением фильтра в предложении WHERE инструкции SELECT, а затем с помощью представления были изменены строки таблицы, то можно изменить некоторое значение так, что задействованная строка уже не будет удовлетворять фильтру предложения WHERE. Возможно даже обновление строк, которые выходят за пределы области фильтра. Предложение WITH CHECK OPTION препятствует подобному исчезновению строк при обновлении через представление, а также ограничивает модификации только строками, которые удовлетворяют критериям фильтра.

При создании представления пользователю не возвращается виртуальная таблица, которую он обычно получает при выполнении инструкции SELECT. Пользователь получит сообщение типа «Представление создано».

Чтобы выполнить представление, т. е. получить данные в виде виртуальной таблицы, необходимо выполнить запрос SELECT к представлению так же, как и к обычной таблице:

```
SELECT * FROM view_name
```

Для удаления представления используется команда DROP VIEW{view [...n]}. За один раз можно удалить несколько представлений.

При выполнении практического задания студенту необходимо для базы данных, структура которой определена в варианте задания, создать три представления с использованием языка SQL. Основные операторы и функции SQL описаны в конспекте лекций и в [1–9].

1.2 Пример выполнения задания

Пусть база данных имеет структуру, представленную информационной моделью на рисунке 1.

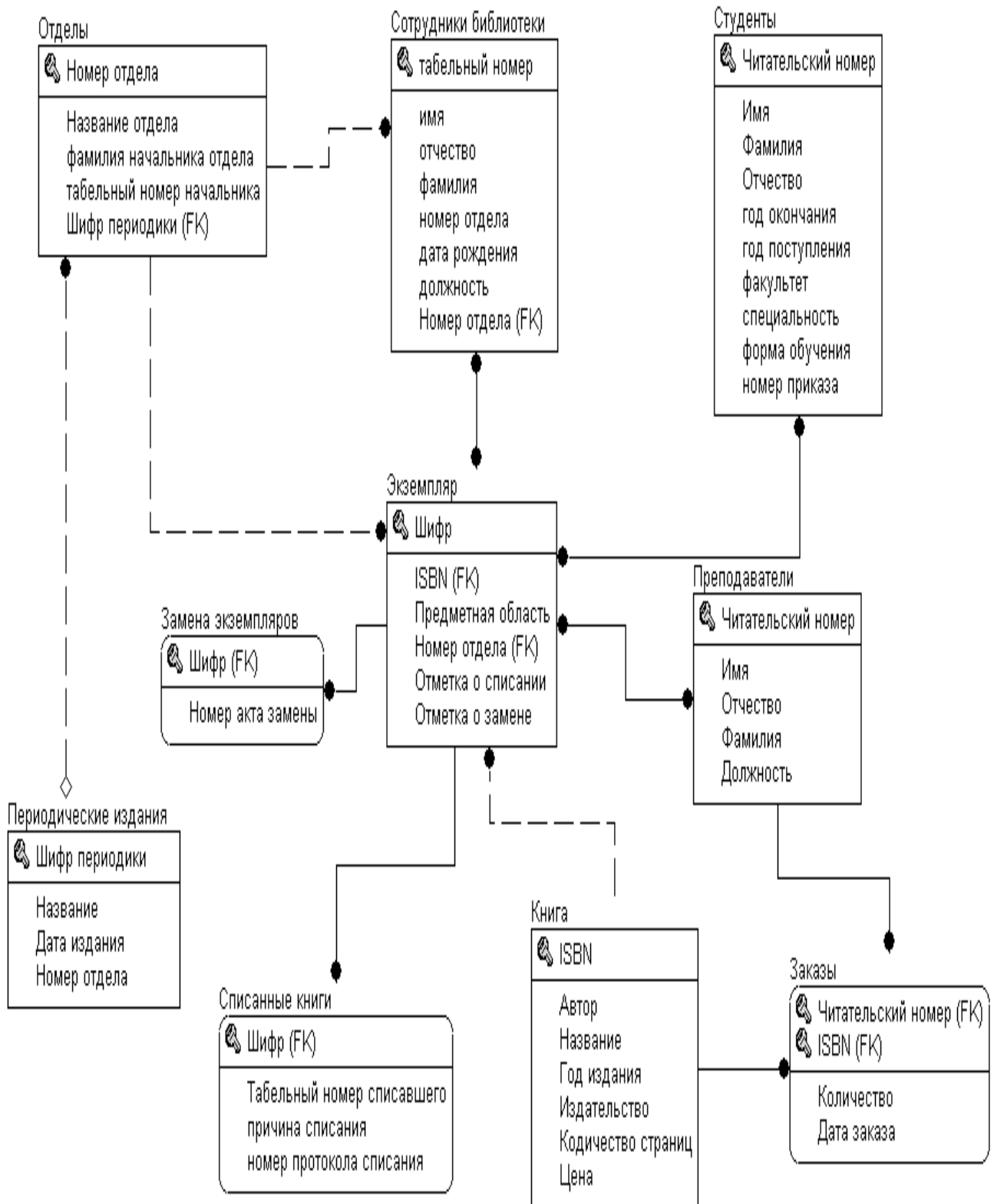


Рисунок 1 – Информационная модель

Разрешение связей «многие-ко-многим», осуществленное на физическом уровне путем введения дополнительных зависимых сущностей «Пользование библиотекой», представлено на рисунке 2.

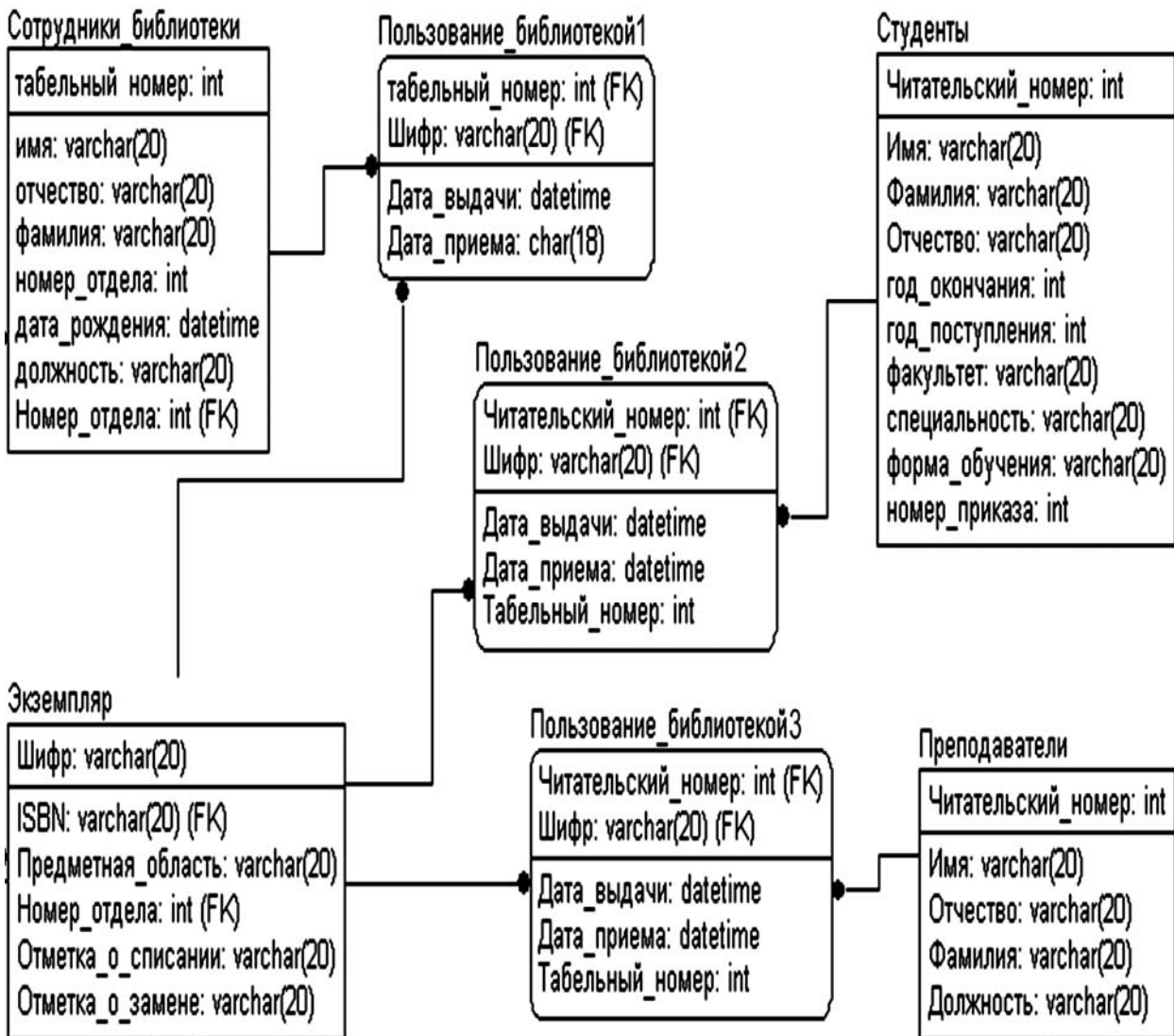


Рисунок 2 – Разрешение связей «многие-ко-многим» на физическом уровне

В базе данных реализованы следующие представления.

1 Вывод ограниченной информации о студентах с фамилией, оканчивающейся на «ов»

```
CREATE VIEW ИнфоСтуденты /* Указываем имя представления */
AS
SELECT /* Указываем, какие поля будут выведены */
    Студенты.Имя, Студенты.Фамилия, Студенты.факультет,
    Студенты.специальность
FROM Студенты /* Из какой таблицы */
WHERE Студенты.Фамилия LIKE '[ ]ов'
```

2 Информация о замене экземпляров по программированию на C++

```
CREATE VIEW ИнфоОЗаменеКопийКниг
AS
SELECT /* Указываем, какие поля из каких таблиц будут выведены */
    Книга.Автор, Книга.Название, Экземпляр.Шифр,
    Замена_экземпляров.Номер_акта_замены
FROM /* Указываем таблицу и связанные с ней при помощи оператора
INNER JOIN левую и правую таблицы, из которых выбираются связанные дан-
ные. После операнда ON указываем, по каким полям связаны две таблицы */
    Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN
    INNER JOIN Замена_экземпляров ON Экземпляр.Шифр =
        Замена_экземпляров.Шифр
WHERE Книга.Название LIKE '%C++' /* Выбираются только те
книги, в названии которых присутствует "C++" */
```

3 Информация о списанных экземплярах по программированию на C++

```
CREATE VIEW ИнфоОСписанииКниг
AS
SELECT /* Указываем, какие поля, из каких таблиц будут выведены */
    Книга.Автор, Книга.Название, Экземпляр.Шифр,
    Списанные_книги.причина_списания,
    Списанные_книги.номер_протокола_списания,
    Списанные_книги.Табельный_номер_списавшего
FROM /* Указываем таблицу, и связанные с ней таблицы, из которых
выбираются связанные данные */
    Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN
    INNER JOIN Списанные_книги ON Экземпляр.Шифр = Списан-
ные_книги.Шифр
WHERE Книга.Название LIKE '%C++' /* Выбираются только те
книги, в названии которых присутствует C++ */
```

4 Информация о преподавателях и их заказах на литературу; количе- ство книг должно быть от 10 до 20 или от 25 до 30

```
CREATE VIEW ИнфоОЗаказах /* Объявляем имя представления */
AS
SELECT /* Указываем, какие поля таблиц будут выведены */
    Преподаватели.Фамилия, Преподаватели.Имя, Преподаватели.Отчество,
    Книга.Автор, Книга.Название, Заказы.Количество
FROM /* Указываем таблицу и связанные с ней таблицы, из которых
выбираются связанные данные*/
    Книга INNER JOIN Заказы ON Книга.ISBN = Заказы.ISBN
    INNER JOIN Преподаватели ON Заказы.Читательский_номер =
        Преподаватели.Читательский_номер
WHERE (Заказы.Количество BETWEEN 10 AND 20) OR
```


(Заказы.Количество BETWEEN 25 AND 30) /* 10 < количество заказов < 20 или 25 < количество заказов < 30 */

5 Информация о книгах, которыми пользуется студент, и табельный номер сотрудника отдела обслуживания, выдавшего книги. Номер отдела обслуживания 11

```
CREATE VIEW ИнфоОКнигахСтудента
AS
SELECT /* Указываем, какие поля таблиц будут выведены */
Студенты.Имя, Студенты.Фамилия, Книга.Автор, Книга.Название,
Сотрудники_библиотеки.Табельный_номер
FROM /* Указываем таблицу, и связанные с ней таблицы, из которых
выбираются связанные данные */
(((Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN)
INNER JOIN Пользование_библиотекой2 ON Экземпляр.Шифр =
Пользование_библиотекой2.Шифр)
INNER JOIN Студенты ON Пользование_библиотекой2.Чита-
тельский_номер = Студенты.Читательский_номер)
INNER JOIN Сотрудники_библиотеки ON Пользование_биб-
лиотекой2.Табельный_номер = Сотрудники_библиотеки.табельный_номер
WHERE Сотрудники_библиотеки.табельный_номер LIKE '11[ ]' /* Та-
бельный номер сотрудников должен начинаться с «11» */
```

6 Информация о книгах, которыми пользуется преподаватель, и табельный номер сотрудника отдела обслуживания, выдавшего книги. Номер отдела обслуживания 11

```
CREATE VIEW ИнфоОКнигахПреподавателя
AS
SELECT /* Указываем, какие поля таблиц будут выведены */
Преподаватели.Имя, Преподаватели.Фамилия, Книга.Автор,
Книга.Название, Сотрудники_библиотеки.Табельный_номер
FROM /* Указываем таблицу, и связанные с ней таблицы, из которых вы-
бираются связанные данные */
(((Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN)
INNER JOIN Пользование_библиотекой3 ON Экземпляр.Шифр = Пользо-
вание_библиотекой3.Шифр)
INNER JOIN Преподаватели ON Пользова-
ние_библиотекой3.Читательский_номер = Преподаватели.Читательский_номер)
INNER JOIN Сотрудники_библиотеки ON Пользование_библио-
текой3.Табельный_номер = Сотрудники_библиотеки.табельный_номер
WHERE Сотрудники_библиотеки.табельный_номер LIKE '11[ ]'
/* Табельный номер сотрудников должен начинаться с 11 */
```

7 Информация об отделах 11 и 12 и работающих в них сотрудниках

```
CREATE VIEW ИнфоОтделы
AS
SELECT /* Указываем, какие поля будут выбраны */
Отделы.номер_отдела, Отделы.название_отдела,
Сотрудники_библиотеки.фамилия, Сотрудники_библиотеки.имя,
Сотрудники_библиотеки.отчество
FROM /* Указываем таблицу и связанные с ней таблицы, из которых
выбираются связанные данные. */
Отделы INNER JOIN Сотрудники_библиотеки ON Отделы.Номер_отдела =
Сотрудники_библиотеки.Номер_отдела
GROUP BY Отделы.Номер.отдела /* По какому полю будут сгруппирова-
ны поля */
WHERE Отделы.Номер_отдела BETWEEN 10 AND 13 /* Выбираются от-
делы, номер которых находится между 10 и 13 */
```

8 Информация о месте хранения периодических изданий, название которых содержит в себе слово «вестник», но при этом должны быть исключены периодические издания, первое слово которых начинается на «Э», например «Экономический вестник»

```
CREATE VIEW ИнфоПериодика
AS
SELECT
Периодические_издания.Название, Периодические_издания.Дата_издания,
Отделы.Название_отдела, Отделы.Номер_отдела
FROM
Отделы INNER JOIN Периодические_издания ON Отделы.Номер_отдела =
Периодические_издания.Номер_отдела
WHERE Периодические_издания.Название LIKE '[^Э] %вестник%'
/* [^Э] запрещает начинаться первому слову на «Э», %вестник% – указы-
вает на то, что оставшаяся часть названия должна содержать слово «вест-
ник» */
```

9 Полная информация об экземплярах, которые были изданы с 2000 до текущего года

```
CREATE VIEW ПолнаяИнфоОбЭкземплярах
AS
SELECT
Экземпляр.шифр, Книга.Автор, Книга.Название, Книга.Год_издания,
Книга.Издательство, Экземпляр.Предметная_область,
Экземпляр.Номер_отдела, Экземпляр.Отметка_о_списании,
Экземпляр.Отметка_о_замене
FROM
Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN
```

```
WHERE Книга.Год_издания BETWEEN 2000 AND YEAR (GETDATE())
/* GETDATE() возвращает текущую дату, YEAR (<дата>) – год <даты> */
```

10 Информация о наличии книги в библиотеке

```
CREATE VIEW ИнфоОНаличииКниги
AS
SELECT
    Книга.ISBN, Экземпляр.Шифр, Экземпляр.Номер_отдела,
    Пользование_библиотекой2.Дата_выдачи,
    Пользование_библиотекой2.Дата_приема
FROM
    (Книга INNER JOIN Экземпляр ON Книга.ISBN = Экземпляр.ISBN)
INNER JOIN
    Пользование_библиотекой2 ON Экземпляр.Шифр = Пользование_библиотекой2.Шифр
WHERE Пользование_библиотекой2.Дата_приема BETWEEN Пользование_библиотекой2.Дата_выдачи AND GETDATE()
/*Если книгу сдали, т. е. она в наличии в библиотеке, то дата приема будет между датой выдачи и текущей датой*/
```

2 Варианты заданий

В вариантах 1–3 база данных имеет структуру, показанную на рисунке 3.

Вариант 1

Реализовать в базе данных (см. рисунок 3) три представления:

- 1) вывести минимальный год поступления групп АСОИЗ в университет;
- 2) вывод списка студентов группы АСОИЗ-201, отсортированного по фамилиям;
- 3) вывести среднее значение идентификатора студентов группы АСОИЗ-201.

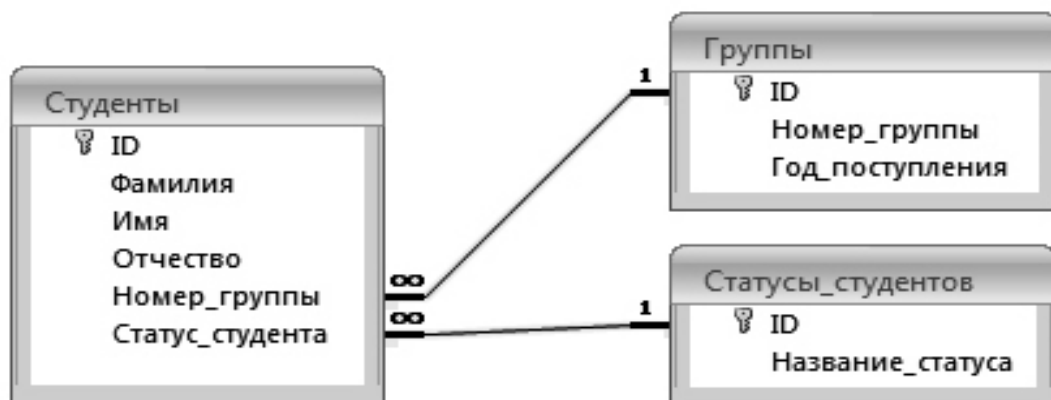


Рисунок 3 – Схема базы данных к вариантам 1–3

Вариант 2

Реализовать в базе данных (см. рисунок 3) три представления:

- 1) вывести список студентов, обучающихся в группе АСОИЗ-201 или АСОИЗ-202;
- 2) вывести количество студентов каждой группы;
- 3) вывести список студентов группы АСОИЗ-201, фамилия которых начинается с буквы «Д».

Вариант 3

Реализовать в базе данных (см. рисунок 3) три представления:

- 1) вывести список студентов, которые числятся в удаленных группах;
- 2) вывести список студентов группы АСОИЗ-201, номера которых в списке с 4 по 10;
- 3) вывести список групп, в которых не учится ни один студент.

В вариантах 4–6 база данных имеет структуру, показанную на рисунке 4.

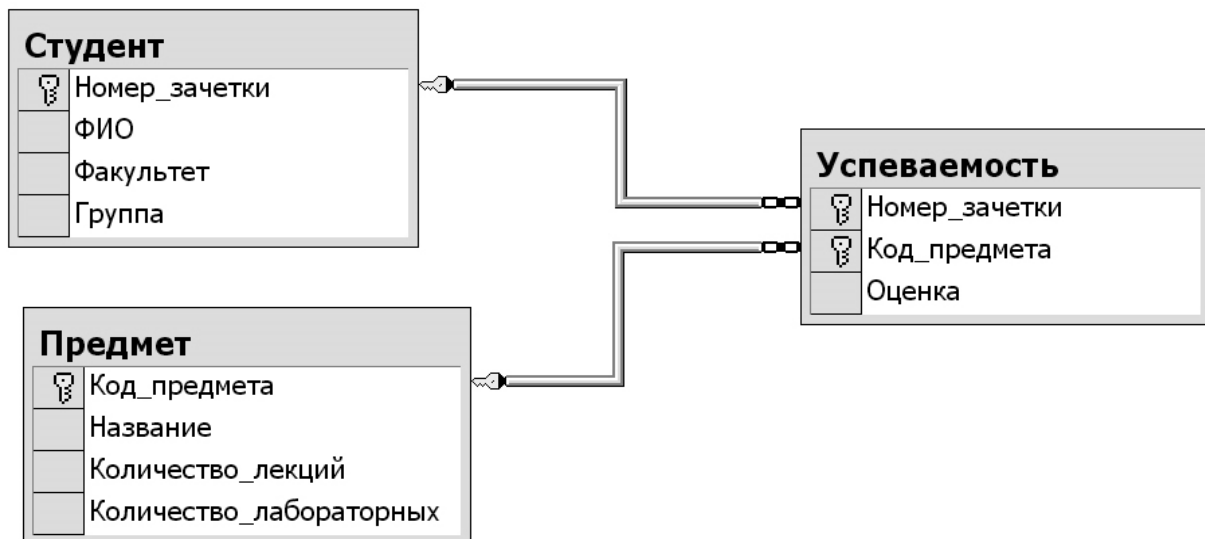


Рисунок 4 – Схема базы данных «Успеваемость студентов»

Вариант 4

Реализовать в базе данных (см. рисунок 4) три представления:

- 1) вычислить средний балл каждого студента;
- 2) выполнить сортировку списка студентов по ФИО;
- 3) подсчитать количество студентов в группе АСОИЗ-201.

Вариант 5

Реализовать в базе данных (см. рисунок 4) три представления:

- 1) вывести максимальное количество лабораторных;
- 2) подсчитать количество предметов, изучаемых группой АСОИЗ-201;
- 3) вывести названия предметов, по которым количество лекций составляет от 20 до 30.

Вариант 6

Реализовать в базе данных (см. рисунок 4) три представления:

- 1) вывести названия предметов в группах, по которым количество лекций меньше 10, а количество лабораторных больше 10;
- 2) вывести названия предметов в группах, по которым количество лекций меньше 20 или количество лабораторных больше 10;
- 3) вывести список студентов, для которых в базе данных отсутствует информация об успеваемости.

В вариантах 7–9 база данных имеет структуру, представленную на рисунке 5.

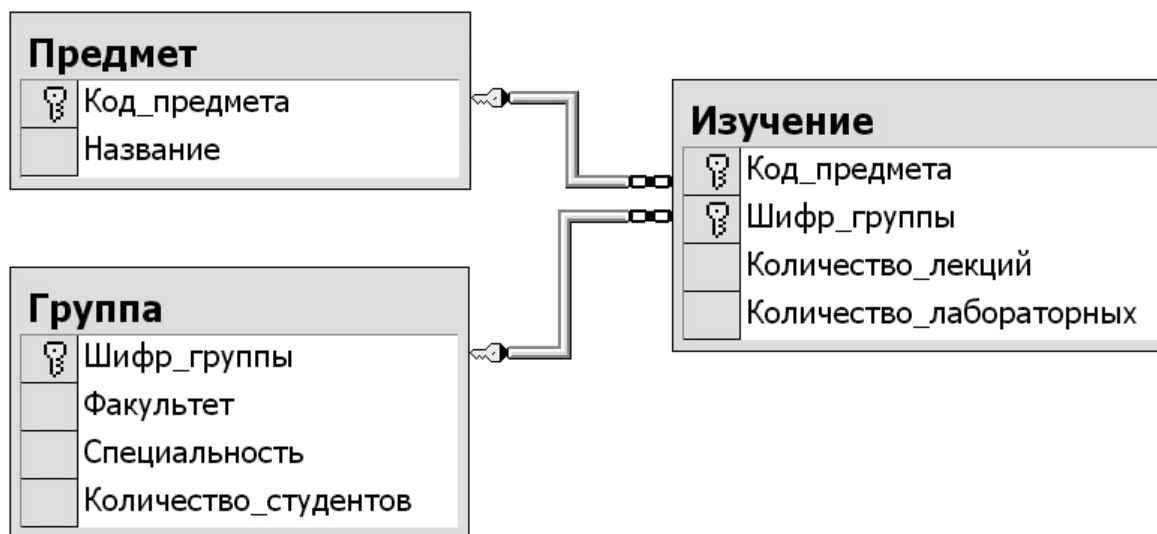


Рисунок 5 – Структура базы данных «Изучение предметов»

Вариант 7

Реализовать в базе данных (см. рисунок 5) три представления:

- 1) вычисление среднего количества лекций по предмету;
- 2) вывести суммарное количество лабораторных по предмету;
- 3) вывести суммарное количество лабораторных в группе АСОИЗ-201.

Вариант 8

Реализовать в базе данных (см. рисунок 5) три представления:

- 1) вывод названия предмета с максимальным количеством лабораторных;
- 2) подсчитать количество предметов, изучаемых группой АСОИЗ-201;
- 3) вывести названия предметов, по которым среднее количество лекций от 30 до 40.

Вариант 9

Реализовать в базе данных (см. рисунок 5) три представления:

- 1) вывести названия предметов в группах, по которым количество лекций меньше 10, а количество лабораторных больше 15;

2) вывести названия предметов в группах, по которым количество лекций меньше 20 или количество лабораторных больше 10;

3) вывести список групп, для которых в базе данных отсутствуют записи в таблице «Изучение».

В вариантах 10–12 база данных имеет структуру, представленную на рисунке 6.

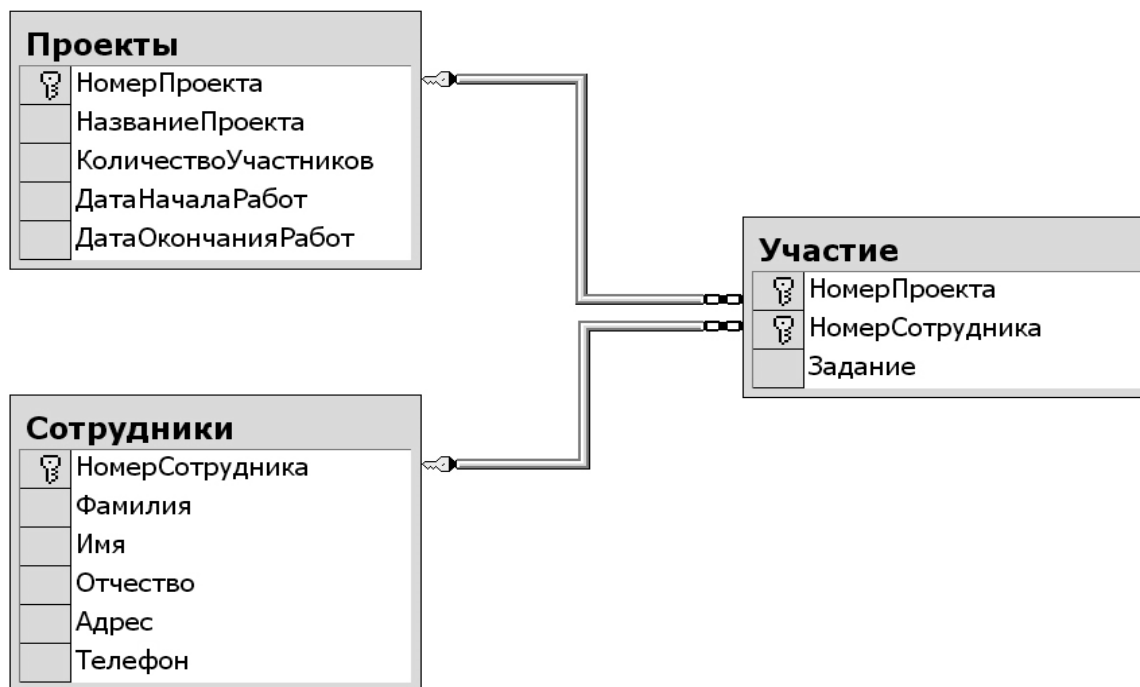


Рисунок 6 – Схема базы данных «Участие сотрудников в проектах»

Вариант 10

Реализовать в базе данных (см. рисунок 6) три представления:

- 1) вывести максимальное количество участников проекта;
- 2) вывести минимальное количество участников проекта;
- 3) подсчитать количество проектов, которые выполняются в настоящее время.

Вариант 11

Реализовать в базе данных (см. рисунок 6) три представления:

- 1) вывести среднее количество участников проектов;
- 2) вывести фамилии, имена и отчества сотрудников, которые участвуют в проекте с номером 5, отсортированные в алфавитном порядке по фамилиям;
- 3) вывести номера и названия проектов, которые выполняются в настоящий момент.

Вариант 12

Реализовать в базе данных (см. рисунок 6) три представления:

- 1) вывести номера и названия проектов, которые уже завершились, а также номера и названия проектов, которые еще не начались;
- 2) вывести номера и названия проектов, в которых заданное количество участников от 3 до 5;
- 3) вывести информацию о проектах, по которым не выдано ни одного задания.

В вариантах 13–15 база данных имеет структуру, представленную на рисунке 7.

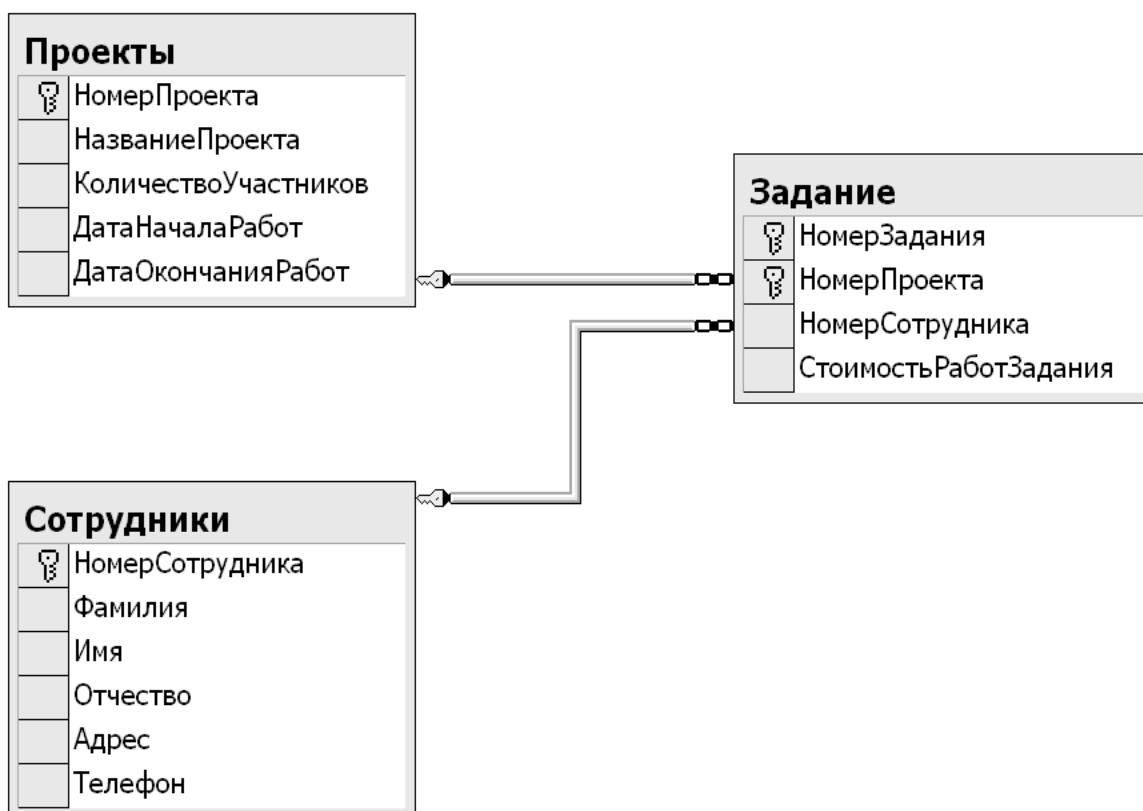


Рисунок 7 – Схема базы данных «Участие сотрудников в проектах»

Вариант 13

Реализовать в базе данных (см. рисунок 7) три представления:

- 1) подсчитать количество сотрудников, участвующих в проекте с номером 5;
- 2) вывести суммарную стоимость работ в проекте с номером 5;
- 3) вывести максимальную стоимость работ задания в проекте с номером 5.

Вариант 14

Реализовать в базе данных (см. рисунок 7) три представления:

- 1) вывести суммарную стоимость работ в проекте с номером 5;

2) вывести номера и названия проектов, которые выполняются в настоящий момент;

3) вывести номера и названия проектов, которые уже завершились, а также номера и названия проектов, которые еще не начались.

Вариант 15

Реализовать в базе данных (см. рисунок 7) три представления:

1) вывести номера и названия проектов, в которых заданное количество участников от 3 до 5;

2) вывести информацию о проектах, по которым не выдано ни одного задания;

3) вывести фамилии, имена и отчества сотрудников, которые участвуют в проекте с номером 5, отсортированные по алфавиту по фамилиям.

Список литературы

1 **Агальцов, В. П.** Базы данных [Электронный ресурс]: учебник: в 2 т. Т. 2: Распределенные и удаленные базы данных / В. П. Агальцов. – Москва: ФОРУМ; ИНФРА-М, 2018. – 271 с. – Режим доступа: <https://znanium.com/catalog/product/1068927>. – Дата доступа: 18.02.2022.

2 **Агальцов, В. П.** Базы данных [Электронный ресурс]: учебник: в 2 кн. Кн. 1: Локальные базы данных / В. П. Агальцов. – Москва: ФОРУМ; ИНФРА-М, 2020. – 352 с.: ил. – Режим доступа: <https://znanium.com/catalog/product/1068927>. – Дата доступа: 18.02.2022.

3 **Бен-Ган, И.** Microsoft SQL Server 2012. Создание запросов: учебный курс Microsoft: пер. с англ. / И. Бен-Ган, Д. Сарка, Р. Талмейдж. – Москва: Русская редакция, 2015. – 720 с. : ил.

4 **Голицына, О. Л.** Базы данных [Электронный ресурс]: учебное пособие / О. Л. Голицына, Н. В. Максимов, И. И. Попов. – 4-е изд., перераб. и доп. – Москва: ФОРУМ; ИНФРА-М, 2020. – 400 с. – Режим доступа: <https://znanium.com/catalog/product/1053934>. – Дата доступа: 18.02.2022.

5 **Дадян, Э. Г.** Данные: хранение и обработка [Электронный ресурс]: учебник / Э. Г. Дадян. – Москва: ИНФРА-М, 2020. – 205 с. – Режим доступа: <https://znanium.com/catalog/product/1045133>. – Дата доступа: 18.02.2022.

6 **Кузин, А. В.** Базы данных: учебное пособие / А. В. Кузин, С. В. Левонисова. – 6-е изд., стер. – Москва: Академия, 2016. – 320 с.

7 **Куликов, С. С.** Реляционные базы данных в примерах [Электронный ресурс]: практическое пособие для программистов и тестировщиков / С. С. Куликов. – Минск: Четыре четверти, 2020. – 424 с. – Режим доступа: http://svyatoslav.biz/relational_databases_book/. – Дата доступа: 18.02.2022.

8 **Куликов, С. С.** Работа с MySQL, MS SQL Server и Oracle в примерах [Электронный ресурс]: практическое пособие / С. С. Куликов. – Минск: БОФФ, 2016. – 556 с. – Режим доступа: http://svyatoslav.biz/database_book/. – Дата доступа: 18.02.2022.

9 **Шустова, Л. И.** Базы данных [Электронный ресурс]: учебник / Л. И. Шустова, О. В. Тараканов. – Москва: ИНФРА-М, 2017. – 304 с. – Режим доступа: www.dx.doi.org/10.12737/11549. – Дата доступа: 18.02.2022.