

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Автоматизированные системы управления»

БАЗЫ ДАННЫХ

*Методические рекомендации к самостоятельной работе
для студентов специальности 6-05-0612-03
«Системы управления информацией»
заочной формы обучения*



УДК 004.65
ББК 32.973.26-018.2
Б17

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Автоматизированные системы управления»
«19» ноября 2024 г., протокол № 4

Составитель канд. техн. наук, доц. Е. М. Борчик

Рецензент канд. техн. наук, доц. В. В. Кутузов

Методические рекомендации к самостоятельной работе по дисциплине
«Базы данных» предназначены для студентов специальности 6-05-0612-03 «Сис-
темы управления информацией» заочной формы обучения.

Учебное издание

БАЗЫ ДАННЫХ

Ответственный за выпуск

А. И. Якимов

Корректор

И. В. Голубцова

Компьютерная верстка

Н. П. Полевнича

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, 2025

Содержание

1 Общие сведения об основных элементах учебного процесса	4
2 Методические указания к выполнению контрольной работы.....	4
3 Варианты заданий.....	11
Список литературы.....	16

1 Общие сведения об основных элементах учебного процесса

Целью учебной дисциплины «Базы и банки данных» является формирование профессиональных компетенций для работы с базами данных в области управления и обработки информации.

Задачи дисциплины:

- изучение технологий создания и работы с базами данных в современных системах управления базами данных (СУБД) и языках программирования;
- изучение технологий для работы с базами данных в сети;
- приобретение навыков по разработке программного интерфейса с современными СУБД.

Дисциплина «Базы и банки данных» студентами заочной формы обучения изучается в 5, 6 и 7 семестрах. Формой контроля знаний по дисциплине в 5 семестре является зачет, в 6 семестре – экзамен, в 7 семестре – курсовой проект. В 6 семестре студенты выполняют одну контрольную работу.

Одной из наиболее эффективных форм получения знаний является самостоятельная работа. В методических рекомендациях содержатся:

- теоретические сведения и примеры решения задач из контрольной работы;
- условия задач для самостоятельного решения в контрольной работе;
- список рекомендуемой литературы [1–13].

2 Методические указания к выполнению контрольной работы

2.1 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 БД. 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, 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.

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

Для создания представлений средствами Transact-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 представление нельзя включать операции вычисления и группировки, т. е. запрещается указание параметров ORDER BY, COMPUTE и COMPUTE BY.

Чтобы выполнить представление, т. е. получить данные в виде виртуаль-

ной таблицы, необходимо выполнить запрос SELECT к представлению так же, как и к обычной таблице:

```
SELECT * FROM view_name
```

Для удаления представления используется команда Transact SQL 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 Transact-SQL. Основные операторы и функции Transact-SQL описаны в конспекте лекций, в данных методических рекомендациях и в [1–13].

2.2 Пример выполнения задания

Пусть база данных имеет структуру, представленную информационной моделью на рисунке 2.1.

Разрешение связей «многие ко многим», осуществленное на физическом уровне путем введения дополнительных зависимых сущностей «Пользование библиотекой», представлено на рисунке 2.2.

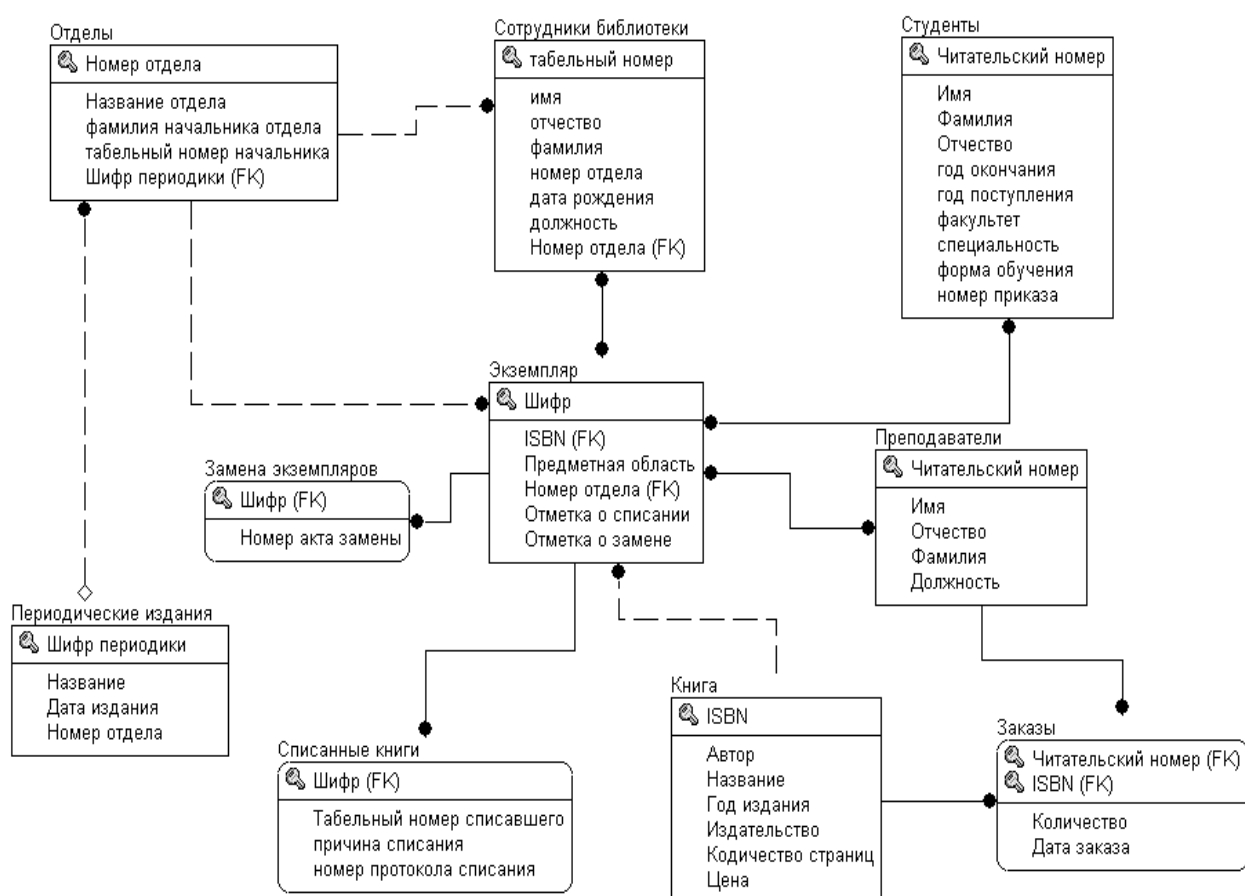


Рисунок 2.1 – Информационная модель

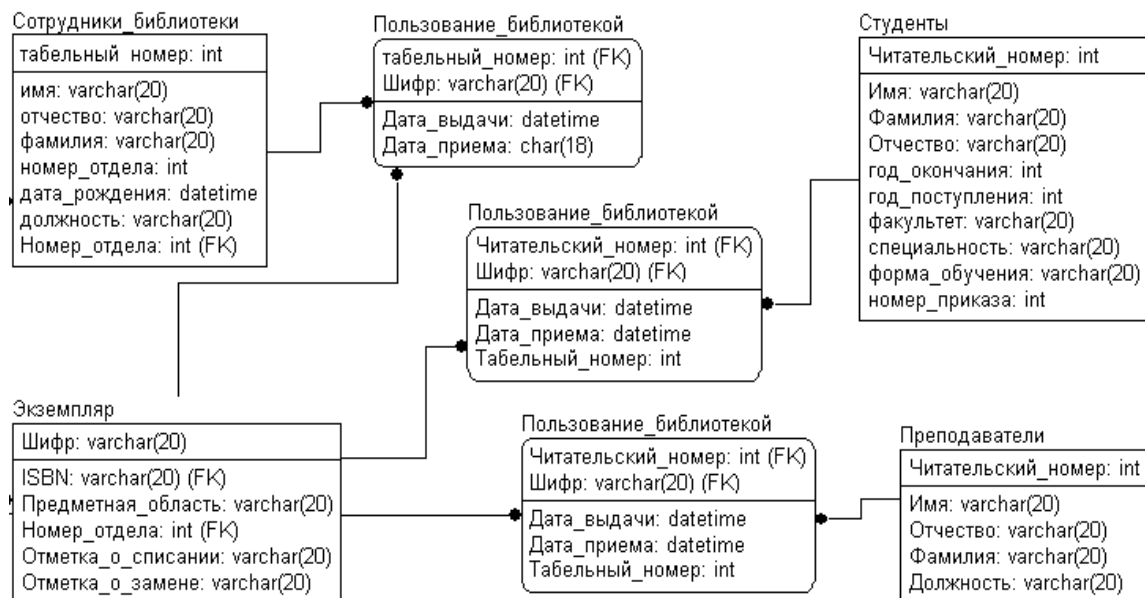


Рисунок 2.2 – Разрешение связей «многие ко многим» на физическом уровне

В базе данных реализованы следующие представления.

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

```

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

2 Информация о замене экземпляров по программированию на C++.

```

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

3 Информация о списанных экземплярах по программированию на Delphi.

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

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

```
CREATE VIEW infoZakazi      /* Объявляем имя представления */
AS
SELECT /* Указываем, какие поля таблиц будут выведены */
Преподаватели.Фамилия, Преподаватели.Имя, Преподаватели.Отчество,
Книга.Автор, Книга.Название, Заказы.Количество
FROM      /* Ука-
зываем таблицу, и связанные с ней таблицы, из которых выбираются связан-
ные данные. */
Книга INNER JOIN (Преподаватели INNER JOIN Заказы ON Преподавате-
ли.Читательский_номер = Заказы.Читательский_номер)
ON Книга.ISBN = Заказы.ISBN
WHERE (Заказы.Количество BETWEEN 10 AND 20) OR (Зака-
зы.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 infoOPolzovaniiStudentov
AS
SELECT /* Указываем, какие поля таблиц будут выведены */
Студенты.Имя, Студенты.Фамилия, Книга.Автор, Книга.Название, Со-
трудники_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 = Пользова-
ние_библиотекой2.Шифр) INNER JOIN
        Студенты ON Пользование_библиотекой2.Читательский_номер = Студен-
ты.Читательский_номер) INNER JOIN
        Сотрудники_библиотеки ON Пользование_библиотекой2.Табель-
ный_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 infoOPolzovaniiPrepodovatelej
AS
SELECT /*Указываем, какие поля таблиц будут выведены*/
        Преподаватели.Имя, Преподаватели.Фамилия, Книга.Автор, Кни-
га.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 = Пользова-
ние_библиотекой3.Шифр) INNER JOIN
        Преподаватели ON Пользование_библиотекой3.Читатель-ский_номер =
Преподаватели.Читательский_номер) INNER JOIN Сотрудники_библиотеки
ON
        Пользование_библиотекой3.Табельный_номер =
Сотрудни-
ки_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 Otdeli
AS
SELECT /* Указываем, какие поля будут выбраны */
        Отделы.номер_отдела, Отделы.название_отдела, Сотрудни-
ки_библиотеки.фамилия, Сотрудники_библиотеки.имя,Сотрудники_биб-
лиотеки.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 номер.отдела /* По какому полю будут сгруппированы 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 INFOPeriodika
AS
SELECT /* Указываем какие поля будут выбраны */
Периодические_издания.Название,          Периодические_издания.Дата_издания,
Отделы.Название_отдела, Отделы.Номер_отдела
FROM /* Указываем таблицу, и связанные с ней таблицы, из которых
выбираются связанные данные. */
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% – указывает на то, что оставшаяся часть названия должна содержать слово «вестник» */
```

9 Полная информация об экземпляре, который был издан с 2000 до текущего года.

```
CREATE VIEW FullInfoEkzemplar
AS
SELECT /* Указываем какие поля будут выбраны */
Экземпляр.шифр, Книга.Автор, Книга.Название, Кни-га.Год_издания,
Книга.Издательство,          Экземпляр.Предметная_область,          Экзем-
пляр.Номер_отдела,          Экземпляр.Отметка_о_списании,          Экзем-
пляр.Отметка_о_замене
FROM /* Указываем таблицу, и связанные с ней таблицы, из ко-
торых выбираются связанные данные. */
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 InfoAboutBook
AS
SELECT /*Указываем какие поля будут выбраны*/
Книга.ISBN, Экземпляр.Шифр, Экземпляр.Номер_отдела, Пользова-
ние_библиотекой2.Дата_выдачи, Пользование_библиотекой2.Дата_приема
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 = Пользова-
ние_библиотекой2.Шифр
WHERE Пользование_библиотекой2.Дата_приема BETWEEN Пользова-
ние_библиотекой2.Дата_выдачи AND GETDATE() /*Если книгу сда-
ли, т. е. она в наличии в библиотеке, то дата приема будет между датой вы-
дачи и текущей датой*/

3 Варианты заданий

В вариантах 1–3 база данных имеет структуру, показанную на рисунке 3.1.

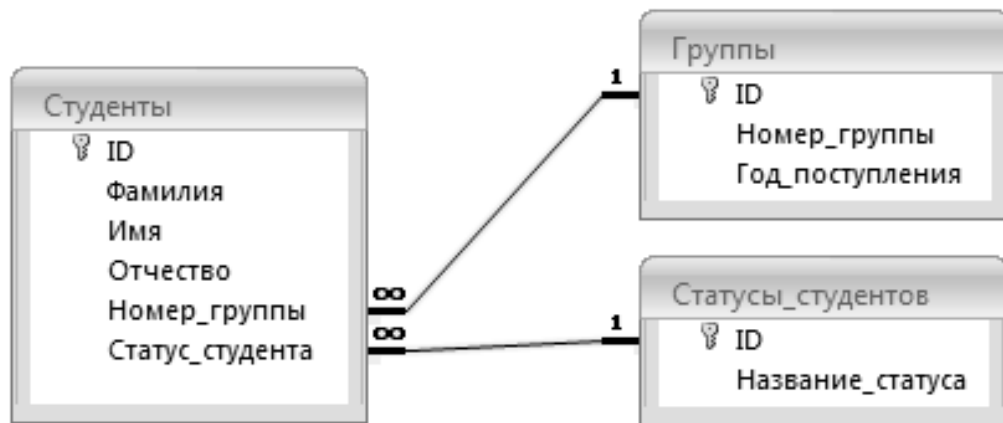


Рисунок 3.1 – Схема базы данных к вариантам 1–3

Вариант 1.

Реализовать в базе данных (см. рисунок 3.1) три представления:

- 1) вывести минимальный год поступления групп АСОИЗ в университет;
- 2) вывести список студентов группы АСОИЗ-041, отсортированный по фамилиям;
- 3) вывести среднее значение идентификатора студентов группы АСОИЗ-041.

Вариант 2.

Реализовать в базе данных (см. рисунок 3.1) три представления:

- 1) вывести список студентов, обучающихся в группе АСОИЗ-041 или АСОИЗ-042;
- 2) вывести количество студентов каждой группы;
- 3) вывести список студентов группы АСОИЗ-041, фамилия которых начинается с буквы «Д».

Вариант 3.

Реализовать в базе данных (см. рисунок 3.1) три представления:

- 1) вывести список студентов, которые числятся в удаленных группах;
- 2) вывести список студентов группы АСОИЗ-041, номера которых в спис-

ке 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 3.2.

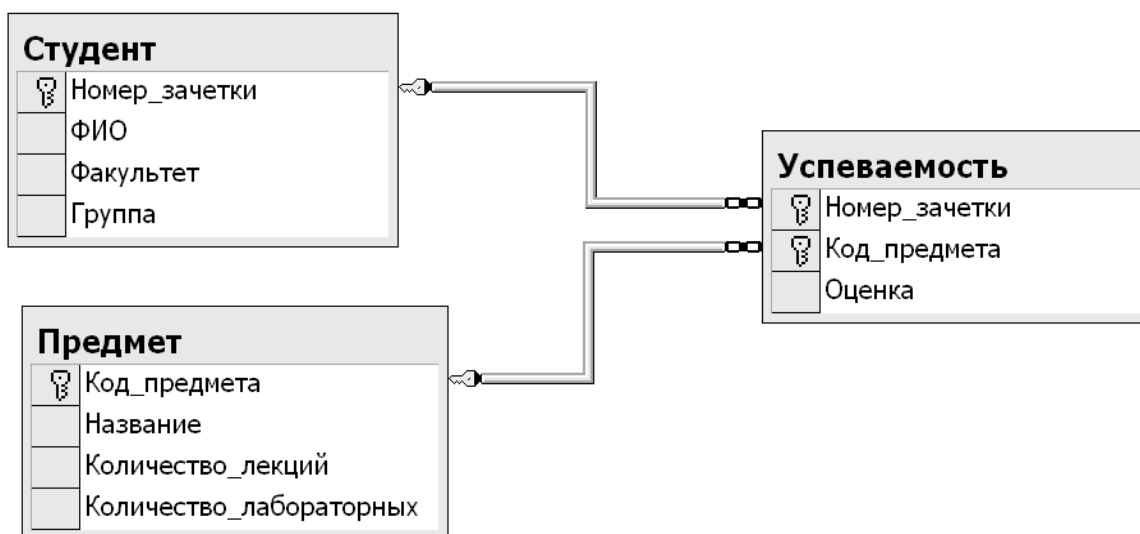


Рисунок 3.2 – Схема базы данных «Успеваемость студентов»

Вариант 4.

Реализовать в базе данных (см. рисунок 3.2) три представления:

- 1) вычислить средний балл каждого студента;
- 2) выполнить сортировку списка студентов по ФИО;
- 3) подсчитать количество студентов в группе АСОИЗ-051.

Вариант 5.

Реализовать в базе данных (см. рисунок 3.2) три представления:

- 1) вывести максимальное количество лабораторных;
- 2) подсчитать количество предметов, изучаемых группой АСОИЗ-051;
- 3) вывести названия предметов, по которым количество лекций от 20 до 30.

Вариант 6.

Реализовать в базе данных (см. рисунок 3.2) 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 3.3.

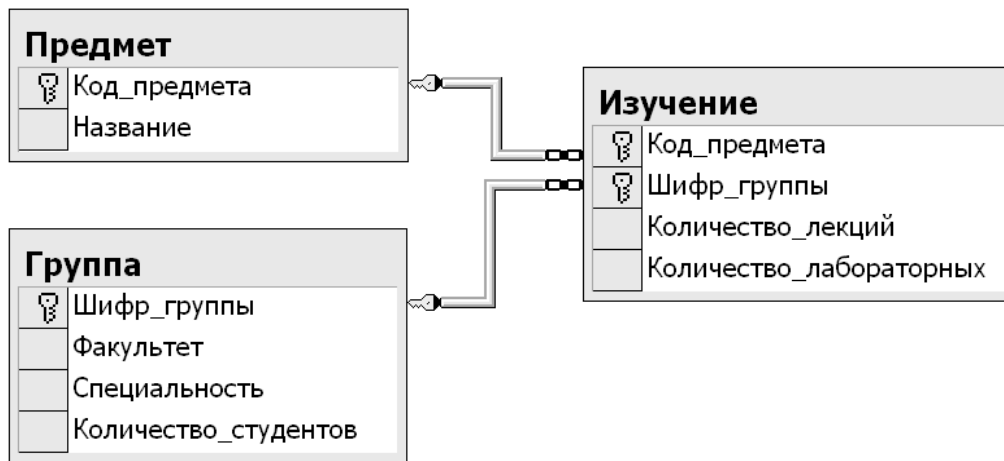


Рисунок 3.3 – Структура базы данных «Изучение предметов»

Вариант 7.

Реализовать в базе данных (см. рисунок 3.3) три представления:

- 1) вычислить среднее количество лекций по предмету;
- 2) вывести суммарное количество лабораторных по предмету;
- 3) вывести суммарное количество лабораторных в группе АСОИЗ-051.

Вариант 8.

Реализовать в базе данных (см. рисунок 3.3) три представления:

- 1) вывести название предмета с максимальным количеством лабораторных;
- 2) подсчитать количество предметов, изучаемых группой АСОИЗ-051;
- 3) вывести названия предметов, по которым среднее количество лекций от 30 до 40.

Вариант 9.

Реализовать в базе данных (см. рисунок 3.3) 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 3.4.

Вариант 10.

Реализовать в базе данных (см. рисунок 3.4) три представления:

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

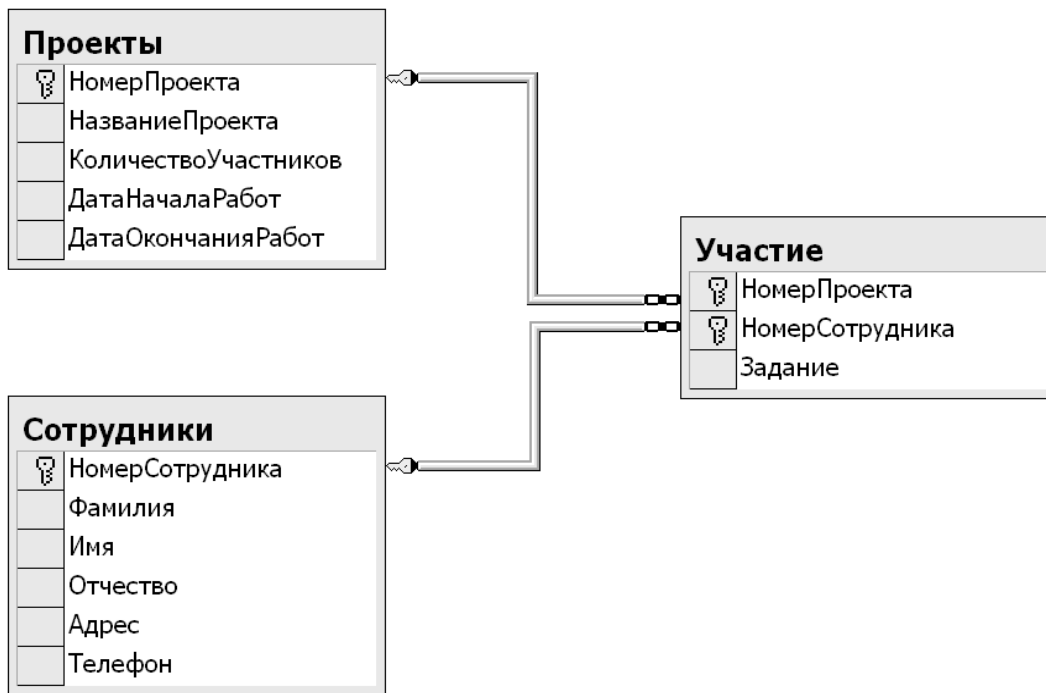


Рисунок 3.4 – Схема базы данных «Участие сотрудников в проектах»

Вариант 11.

Реализовать в базе данных (см. рисунок 3.4) 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 3.4) 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 3.5.

Вариант 13.

Реализовать в базе данных (см. рисунок 3.5) 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.

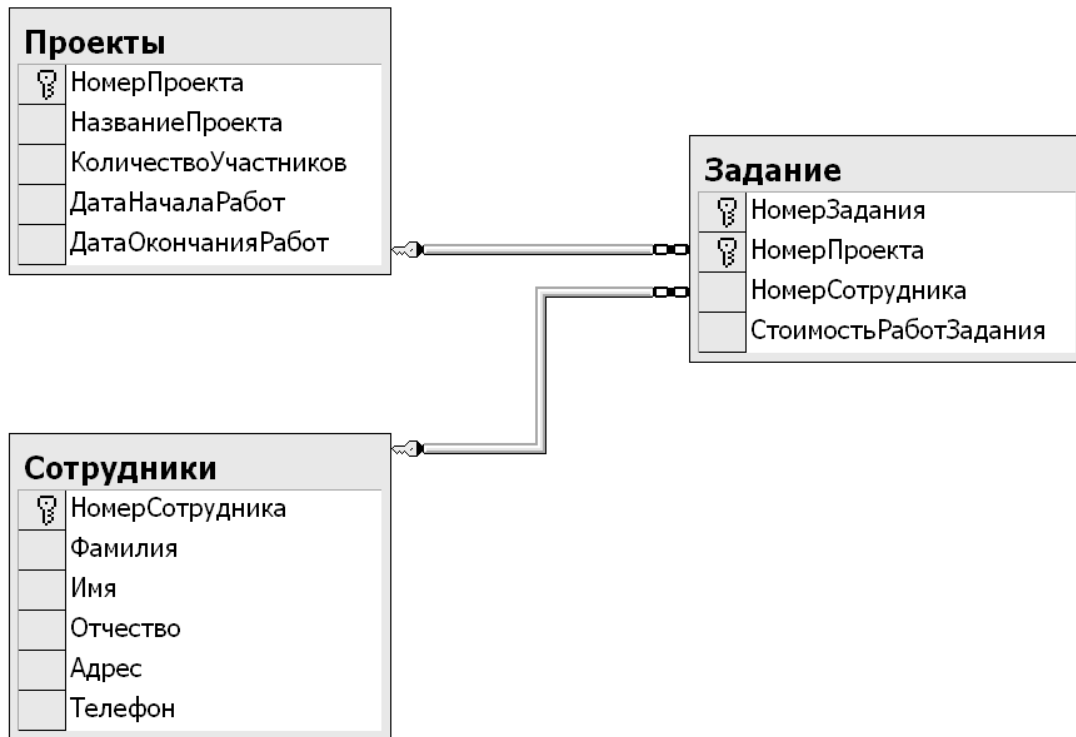


Рисунок 3.5 – Схема базы данных «Участие сотрудников в проектах»

Вариант 14.

Реализовать в базе данных (см. рисунок 3.5) 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 3.5) 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: в 2 т. Т. 2: Распределенные и удаленные базы данных : учебник / В. П. Агальцов. – М. : ФОРУМ; ИНФРА-М, 2015. – 272 с.: ил.
- 2 **Бен-Ган, И.** Microsoft SQL Server 2012. Создание запросов : учеб. курс Microsoft : пер. с англ. / И. Бен-Ган, Д. Сарка, Р. Талмейдж. – М. : Рус. редакция, 2015. – 720 с. : ил.
- 3 **Виейра, Р.** Программирование баз данных Microsoft SQL Server 2005 для профессионалов: пер. с англ. / Р. Виейра. – М. : Диалектика, 2008. – 1072 с.
- 4 **Дейт, К.** Введение в системы баз данных / К. Дейт. – 6-е изд., перераб. и доп. – К. : Диалектика, 2004. – 843 с.
- 5 **Жилинский, А. А.** Самоучитель Microsoft SQL Server 2005 / А. А. Жилинский. – СПб. : БХВ-Петербург, 2007. – 224 с.
- 6 **Кузин, А. В.** Базы данных : учеб. пособие / А. В. Кузин, С. В. Левонисова. – 6-е изд., стер. – М. : Академия, 2016. – 320 с.
- 7 **Мамаев, Е. В.** Microsoft SQL Server 2000 / Е. В. Мамаев. – СПб. : Питер, 2002. – 460 с.
- 8 **Мейер, Д.** Теория реляционных баз данных / Д. Мейер. – М. : Финансы и статистика, 2003. – 560 с.
- 9 **Орин, Т.** Оптимизация и администрирование баз данных Microsoft SQL Server 2005. Учебный курс Microsoft : пер. с англ. / Т. Орин, М. Йен. – М. : Рус. редакция, 2007. – 624 с.
- 10 **Риордан, Р. М.** Программирование в Microsoft SQL Server 2000. Шаг за шагом / Р. М. Риордан. – М. : Финансы и статистика, 2002. – 540 с.
- 11 **Советов, Б. Я.** Базы данных: теория и практика : учебник для бакалавров. – 2-е изд. – М. : Юрайт, 2012. – 463 с.
- 12 **Ульман, Дж.** Основы систем баз данных / Дж. Ульман. – М. : Финансы и статистика, 2002. – 304 с.
- 13 **Шустова, Л. И.** Базы данных : учебник / Л. И. Шустова, О. В. Тараканов. – М.: ИНФРА-М, 2017. – 304 с.