

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
РЯЗАНСКИЙ ГОСУДАРСТВЕННЫЙ РАДИОТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ ИМЕНИ В.Ф. УТКИНА

КАФЕДРА ЭВМ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ
ПО ДИСЦИПЛИНЕ «БАЗЫ ДАННЫХ»

СОДЕРЖАНИЕ

1	Семантическое описание предметной области.....	3
2	Разработка базы данных.....	4
2.1	Разработка логической модели данных	4
2.2	Определение типов хранимых данных.....	8
2.3	Задание ограничений целостности данных	9
2.4	Построение схемы БД	10
2.5	Разработка сценария для создания БД и основных объектов структуры БД	11
3	Разработка основных операций реляционной алгебры	14
4	Разработка запросов к БД для реализации информационных потребностей пользователя	18

Пример выполнения курсовой работы по дисциплине «Базы данных»

Вариант задания.

База данные «Сессия» должна хранить информацию о сдаче студентами экзаменов в текущую сессию. В базе требуется хранить данные:

- 1) О студентах, а именно:
номер зачётной книжки, ФИО студента, номер группы, номер телефона, дата рождения.
- 2) О преподавателях, а именно:
ФИО преподавателя, кафедра, должность, ученая степень, стаж работы, оклад.
- 3) О дисциплинах, проводимых преподавателями, с указанием кода и наименования дисциплины, а также общего количества часов, отведенных на дисциплину.
- 4) О сдаче студентом экзамена по дисциплине с указанием даты сдачи экзамена и полученной оценки.

1 Семантическое описание предметной области

Сессией называется период сдачи экзаменов по дисциплинам, которые преподавались на протяжении учебного семестра.

Каждый студент должен сдавать определенное количество экзаменов по дисциплинам в соответствии с учебным планом. О студенте требуется хранить следующую информацию: номер зачётной книжки, ФИО студента, номер группы, номер телефона, дата рождения. Могут быть записи о студентах, которые пока не сдали ни одного экзамена.

Экзамен принимается преподавателем у целой группы, а, значит, экзамен по одной и той же дисциплине сдает определенное количество студентов. Оценки по экзамену представлены следующим множеством чисел: {2, 3, 4, 5}. О каждом экзамене определенного студента по заданному предмету требуется хранить следующую информацию: дату сдачи экзамена и полученную оценку. Студент может сдавать экзамен по одной и той же дисциплине несколько раз.

О дисциплинах требуется хранить следующую информацию: код и наименование дисциплины, а также общее количество часов, отведенных на дисциплину. Могут быть записи о дисциплинах, которые пока не сдал ни один студент.

Каждая дисциплина должна быть закреплена за определенным преподавателем. О преподавателях требуется хранить следующую информацию: ФИО преподавателя, кафедра, должность, ученую степень, стаж работы, оклад. Причем преподаватель может вести и, соответственно, принимать экзамен по нескольким дисциплинам. Но за каждой дисциплиной должен быть закреплён только один преподаватель, принимающий экзамен.

2 Разработка базы данных

2.1 Разработка логической модели данных

Описание основных сущностей ПО.

На основании проведенного анализа предметной области «Сессия» можно выделить следующие сущности:

Студент, Преподаватель, Дисциплина, Экзамен.

Представим список сущностей предметной области (таблица 1).

Таблица 1. Список сущностей

№	Название	Назначение
1	Студент	Перечень студентов, обучающихся в университете
2	Преподаватель	Описание профессорско-преподавательского состава университета
3	Дисциплина	Перечень дисциплин, по которым студенты обязаны сдавать экзамены в соответствии с учебным планом
4	Экзамен	Экзаменационные ведомости с перечнем студентов и их оценками

Для каждой сущности необходимо привести описание ее атрибутов в виде таблицы. На данном этапе выделяются ключевые (это первичные, внешние и альтернативные ключи) и неключевые атрибуты. Результаты приведены в таблицах 2-5.

Следует отметить, что для сущности Преподаватель, в отличие от других сущностей, нет атрибута, с помощью которого можно идентифицировать конкретного преподавателя. Поэтому требуется ввести дополнительный атрибут, например, Табельный номер, который и будет являться первичным ключом.

Атрибут Табельный номер является дополнительным служебным полем, добавленным к уже имеющимся информационным полям таблицы. Такие атрибуты называются суррогатными, их единственное предназначение - служить первичным ключом. Обычно значение таких полей генерируется СУБД искусственно с помощью специальных типов. При добавлении новой записи в таблицу автоматически записывается уникальное для этой таблицы числовое значение — так называемый «автоинкремент» (в Microsoft SQL Server – свойство IDENTITY). Именно его и нужно использовать для автогенерации значений атрибута Табельный номер. Можно указать начальное значение и значение приращения, которое прибавляется к предыдущему значению на каждом шаге.

Таблица 2. Список атрибутов таблицы «Преподаватель»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Табельный номер	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому преподавателю. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
	ФИО	
	Кафедра	
	Должность	
	Ученая степень	
	Стаж работы	
	Оклад	

Таблица 3. Список атрибутов таблицы «Студент»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Номер зачётной книжки	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому студенту. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
	ФИО	
	Номер группы	
	Номер телефона	
	Дата рождения	

Таблица 4. Список атрибутов таблицы «Дисциплина»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код дисциплины	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждой дисциплине. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
	Наименование	
ВК (внешний ключ)	Преподаватель	С помощью данного внешнего ключа можно определить, какой преподаватель ведёт данную дисциплину.
	Количество часов	

В таблице Сессия отсутствует уникальное поле, которое можно было бы использовать в качестве первичного ключа. Исходя из предметной области, в один день студент может сдать только один экзамен, то есть сочетание значений студента и даты должно быть уникальным. Поэтому в данной таблице имеет место быть составной первичный ключ (Студент, Дата сдачи экзамена).

Таблица 5. Список атрибутов таблицы «Сессия»

Ключевое поле	Название	Назначение
ПК (первичный ключ) ВК (внешний ключ)	Студент	Является частью составного первичного ключа. С помощью данного внешнего ключа можно определить, какой студент сдаёт экзамен.
ВК (внешний ключ)	Дисциплина	С помощью данного внешнего ключа можно определить, какую дисциплину сдают на экзамене.
ПК (первичный ключ)	Дата сдачи экзамена	Является частью составного первичного ключа.
	Оценка	

Построение логической модели

На основании семантического описания предметной области и списка атрибутов из таблиц 2-5 опишем классы объектов (сущностей) и их свойства, расставим существующие связи между ними и приведем обоснование типов связей (1:1, 1:M и т.д.).

Результат требуется представить в виде таблицы Список связей с описанием характеристик связей (таблица 6) и логической модели в графическом виде, где будут изображены все таблицы и связи между ними (рисунок 1).

Таблица 6. Список связей

№	Сущности, участвующие в связи	Тип связи	Обоснование
1	Преподаватель - Дисциплина	1:N	Преподаватель может вести несколько дисциплин одновременно, но конкретная дисциплина должна быть закреплена только за одним преподавателем.
2	Дисциплина - Экзамен	1:N	Экзамен по одной и той же дисциплине могут сдавать несколько студентов, но запись о конкретной сдаче экзамена может относиться только к одной дисциплине.
3	Студент - Экзамен	1:N	Каждый студент может сдавать несколько экзаменов, но запись о конкретной сдаче экзамена может относиться только к одному студенту.

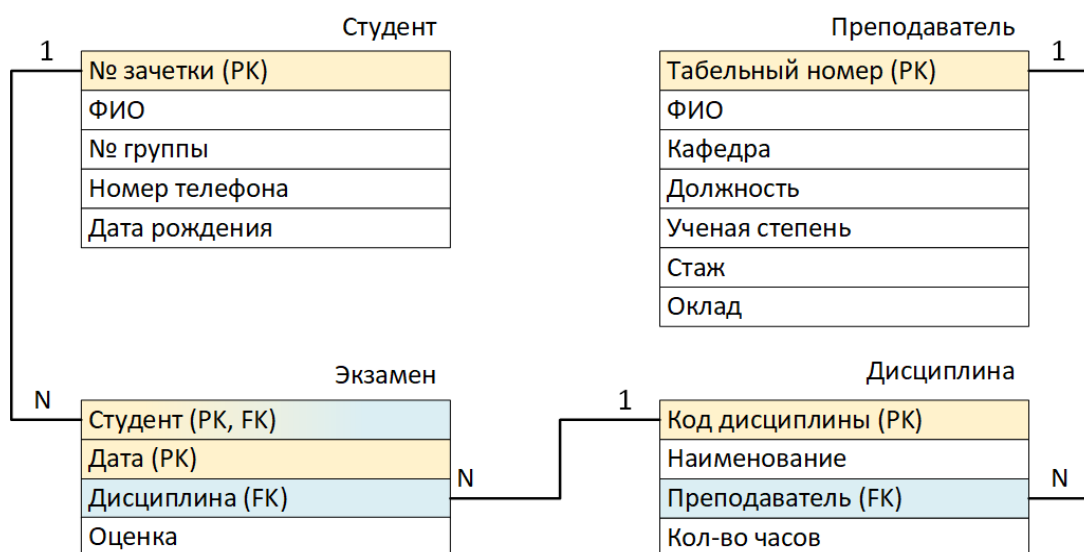


Рисунок 1 – Логическая модель предметной области «Сессия»

2.2 Определение типов хранимых данных

Для каждой таблицы приведем список всех атрибутов с указанием типа данных. При выборе типов данных следует учитывать, что далее для реализации будет использоваться СУБД Microsoft SQL Server. Результаты приведены в таблицах 6-9.

Таблица 6. Список атрибутов таблицы «Студент»

Название	Тип данных
Номер зачётной книжки	int
ФИО	varchar(30)
Номер группы	varchar(5)
Номер телефона	varchar(18)
Дата рождения	date

Таблица 7. Список атрибутов таблицы «Преподаватель»

Название	Тип данных
Табельный номер	int, нач. значение = 1, приращение = 1
ФИО	varchar(30)
Кафедра	varchar(10)
Должность	varchar(21)
Ученая степень	varchar(6)
Стаж работы	int
Оклад	money

Таблица 8. Список атрибутов таблицы «Дисциплина»

Название	Тип данных
Код дисциплины	int
Наименование	varchar(30)
Преподаватель	int
Количество часов	int

Таблица 9. Список атрибутов таблицы «Сессия»

Название	Тип данных
Студент	int
Дисциплина	int
Дата сдачи экзамена	date
Оценка	int

2.3 Задание ограничений целостности данных

Для каждой таблицы необходимо выделить ограничения, накладываемые на значения атрибутов. Это могут быть ограничения на значения, значения по умолчанию, запрет/разрешение использования NULL-значений, поддержка уникальности значений и др. Результаты представлены в таблицах 10-13.

Таблица 10. Ограничения таблицы «Студент»

Название	Ограничение
Номер зачётной книжки	Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1.
ФИО	
Номер группы	NNNL1L2, где N – цифра от 0 до 9; L1 – показывает уровень образования, может принимать следующие значения: Б – бакалавр, С – специалист, М – магистр; L2 – показывает форму обучения, может принимать следующие значения: О – очное, З – заочное, В – очно-заочное (вечернее) обучение. Например, 945БЗ.
Номер телефона	+7 (NNN) NNN-NN-NN, где N – цифра от 0 до 9.
Дата рождения	ДД-ММ-ГГГГ.

Таблица 11. Ограничения таблицы «Преподаватель»

Название	Ограничение
Табельный номер	Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1.
ФИО	
Кафедра	
Должность	Ассистент, старший преподаватель, доцент, профессор. Значение по умолчанию – доцент.

Продолжение таблицы 11

Ученая степень	Д.т.н., к.т.н. Значение по умолчанию – к.т.н.
Стаж работы	Значения от 0 до 75.
Оклад	Значение по умолчанию – 11 000 рублей.

Таблица 12. Ограничения таблицы «Дисциплина»

Название	Ограничение
Код дисциплины	Первичный ключ. Это уникальное значение, не может принимать NULL-значения. Значения автоматически генерируются СУБД, начиная от 1 с шагом 1.
Наименование	
Преподаватель	Внешний ключ.
Количество часов	Число от 16 до 256.

Таблица 13. Ограничения таблицы «Сессия»

Название	Назначение
Студент	Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ.
Дисциплина	Атрибут, входящий в составной первичный ключ. Атрибут, входящий во внешний ключ.
Дата экзамена	Атрибут, входящий в составной первичный ключ. ДД-ММ-ГГГГ
Оценка	5, 4, 3, 2.

2.4 Построение схемы БД

Схема БД представляет собой набор заголовков таблиц. То есть на схеме БД указываются имена таблиц (сущностей), их атрибуты и связи между таблицами. В нашем случае схема связей представлена на рисунке.

Для уточнения схемы БД для каждой сущности (таблицы) указывают наименование атрибута, идентификатор, тип, длина, формат, ограничения. На рисунке 2 представлена схема базы данных «Сессия».

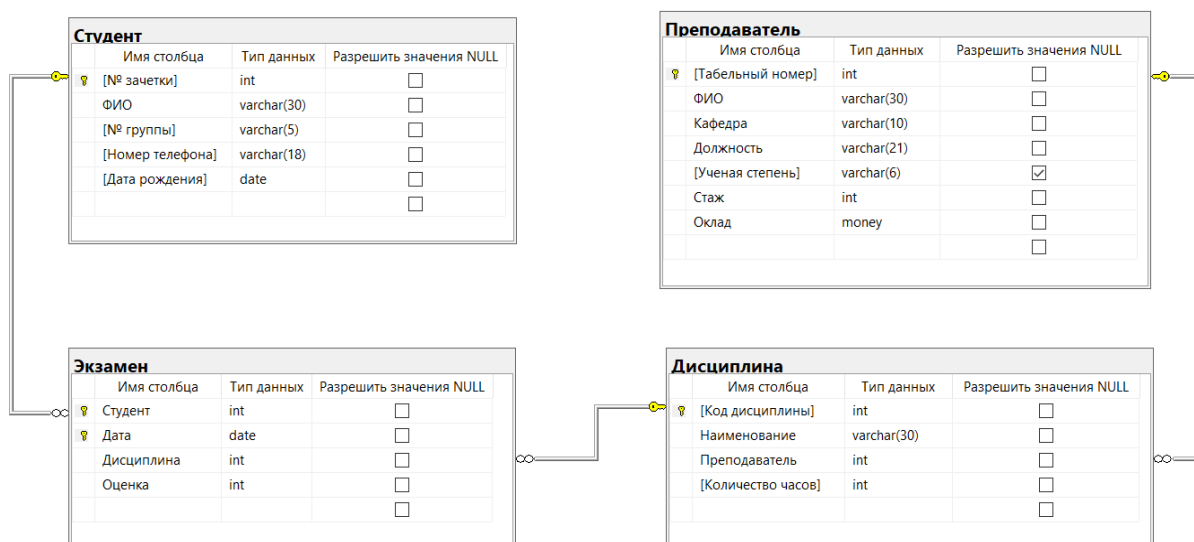


Рисунок 2 – Схема базы данных «Сессия»

2.5 Разработка сценария для создания БД и основных объектов структуры БД

В данном разделе требуется описать сценарий создания разработанной БД и основных объектов структуры БД. При создании таблиц будем использовать следующие способы задания ограничений:

1. При описании оператора CREATE TABLE будем задавать следующие ограничения:

- ограничения NOT NULL как ограничение на столбец;
- ограничения DEFAULT как ограничение на столбец;
- неповторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться только один раз в БД. Такие ограничения будем задавать как ограничения на столбец или на таблицу (в зависимости от количества столбцов, входящих в ограничение).

2. При описании оператора ALTER TABLE будем задавать следующие ограничения:

- повторяющиеся ограничения CHECK и UNIQUE, то есть, если конкретное ограничение будет использоваться несколько раз в БД;
- первичные ключи (простые и составные);
- внешние ключи.

Всем ограничениям, описанным в операторе ALTER TABLE, будем задавать имена.

--Создание файлов БД

```
USE master
```

```
CREATE DATABASE Сессия
ON
```

```
(NAME = 'Сессия',
FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Сессия.mdf',
```

```

        SIZE = 1,
        MAXSIZE = 10,
        FILEGROWTH = 1)
LOG ON
    (NAME = 'Сессия_log',
     FILENAME = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\DATA\Сессия_log.ldf',
     SIZE = 1,
     MAXSIZE = 5,
     FILEGROWTH = 1)
GO

```

--Создание и заполнение таблиц

```

USE Сессия
GO

```

```

CREATE TABLE Студент
([№ зачетки] int not null identity (1, 1),
ФИО varchar(30) not null,
[№ группы] varchar(5) not null check ([№ группы] like '[0-9][0-9][0-9][БСМ][03В]'),
[Номер телефона] varchar(18) not null check ([Номер телефона] like '+7 ([0-9][0-9][0-9])
[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),
[Дата рождения] date not null)
GO

```

```

ALTER TABLE Студент
    ADD CONSTRAINT PK_Студент primary key ([№ зачетки])
GO

```

```

CREATE TABLE Преподаватель
([Табельный номер] int not null identity (1, 1),
ФИО varchar(30) not null,
Кафедра varchar(10) not null,
Должность varchar(21) not null check (Должность in ('Ассистент', 'Старший преподаватель',
'Доцент', 'Профессор')),
[Ученая степень] varchar(6) null check ([Ученая степень] in ('д.т.н.', 'к.т.н.)) default
'к.т.н.',
Стаж int not null check (Стаж between 0 and 75),
Оклад money not null check (Оклад > 0) default 11000)
GO

```

```

ALTER TABLE Преподаватель
    ADD CONSTRAINT PK_Преподаватель primary key ([Табельный номер])
GO

```

```

CREATE TABLE Дисциплина
([Код дисциплины] int not null identity (1, 1),
Наименование varchar(30) not null,
Преподаватель int not null,
[Количество часов] int not null check ([Количество часов] between 16 and 256))
GO

```

```

ALTER TABLE Дисциплина
    ADD CONSTRAINT PK_Дисциплина primary key ([Код дисциплины])
GO
ALTER TABLE Дисциплина
    ADD CONSTRAINT FK_Дисциплина_Преподаватель foreign key (Преподаватель) references
Преподаватель([Табельный номер])
GO

```

```

CREATE TABLE Экзамен
(Студент int not null,
Дата date not null,

```

```

Дисциплина int not null,
Оценка int not null check (Оценка in (2, 3, 4, 5)))
GO

ALTER TABLE Экзамен
    ADD CONSTRAINT PK_Экзамен primary key (Студент, Дата)
GO
ALTER TABLE Экзамен
    ADD CONSTRAINT FK_Экзамен_Студент foreign key (Студент) references Студент([№
зачетки])
GO
ALTER TABLE Экзамен
    ADD CONSTRAINT FK_Экзамен_Дисциплина foreign key (Дисциплина) references
Дисциплина([Код дисциплины])
GO

DBCC CHECKIDENT('Студент', RESEED, 0) --сброс счетчика
GO
INSERT INTO Студент VALUES
    ('Дьяченко Андрей Витальевич', '94050', '+7 (920) 384-67-29',
    '2001-04-12'),
    ('Качанов Никита Васильевич', '94050', '+7 (999) 283-48-27',
    '2000-07-15'),
    ('Краснобаев Дмитрий Сергеевич', '94050', '+7 (910) 947-00-25',
    '2002-11-24'),
    ('Михалин Бегли Владимирович', '946M0', '+7 (910) 572-64-19',
    '2001-12-13'),
    ('Поздняков Кирилл Николаевич', '94550', '+7 (915) 365-07-38',
    '2001-06-20'),
    ('Сапаров Никита Алексеевич', '94550', '+7 (920) 395-78-18',
    '2001-11-09'),
    ('Усманов Антон Михайлович', '94550', '+7 (915) 194-69-12',
    '2002-12-21'),
    ('Чалов Виктор Александрович', '04550', '+7 (910) 947-18-05',
    '2001-01-09'),
    ('Шашков Руслан Сергеевич', '04550', '+7 (920) 006-18-38',
    '2002-08-22'),
    ('Щавелев Егор Вадимович', '04050', '+7 (920) 038-17-34',
    '2002-09-17')
GO

DBCC CHECKIDENT('Преподаватель', RESEED, 0) --сброс счетчика
GO
INSERT INTO Преподаватель VALUES
    ('Гринченко Наталья Николаевна', 'ЭВМ', 'Доцент',
    'к.т.н.', 25, 50000),
    ('Ефимов Алексей Игоревич', 'ЭВМ', 'Доцент',
    'к.т.н.', 6, 50000),
    ('Хизриева Наталья Илесовна', 'ЭВМ', 'Ассистент',
    null, 2, 20000),
    ('Дудко Илья Сергеевич', 'ЭВМ', 'Старший
преподаватель', null, 5, 30000),
    ('Грушин Владимир Александрович', 'САПР ВС', 'Доцент',
    'к.т.н.', 15, 50000),
    ('Колокольчикова Нина Олеговна', 'САПР ВС', 'Профессор',
    'д.т.н.', 40, 75000),
    ('Горкина Ирина Евгеньевна', 'ВПМ', 'Доцент',
    'к.т.н.', 20, 50000),
    ('Заруцкий Иван Степанович', 'КТ', 'Старший
преподаватель', null, 6, 30000),
    ('Капустин Михаил Вадимович', 'КТ', 'Профессор',
    'д.т.н.', 35, 75000),
    ('Птичкин Яков Русланович', 'ИБ', 'Доцент',
    'к.т.н.', 10, 50000)

```

GO

DBCC CHECKIDENT('Дисциплина', RESEED, 0) --сброс счетчика

GO

```
INSERT INTO Дисциплина VALUES
('Базы данных', 1, 64),
('Техническое документирование', 2, 36),
('Проектирование моделей данных', 1, 36),
('Информатика', 4, 64),
('Программирование', 7, 72),
('Моделирование САПР', 5, 36),
('Электротехника и электроника', 6, 64),
('Основы сетевых технологий', 7, 72),
('Информационная безопасность', 10, 36),
('Компьютерная графика', 9, 64)
```

GO

```
INSERT INTO Экзамен VALUES
(1, '2020-06-15', 1, 5),
(1, '2020-06-18', 2, 3),
(1, '2020-06-29', 2, 4),
(1, '2020-06-22', 3, 5),
(2, '2020-06-15', 4, 2),
(2, '2020-06-29', 4, 2),
(2, '2020-06-30', 4, 3),
(2, '2020-06-18', 5, 4),
(3, '2020-06-15', 5, 5),
(3, '2020-06-18', 6, 5),
(3, '2020-06-22', 8, 5),
(3, '2020-06-25', 10, 5),
(4, '2020-06-15', 5, 4),
(4, '2020-06-18', 10, 5),
(5, '2020-06-15', 7, 2),
(5, '2020-06-29', 7, 4),
(6, '2020-06-15', 5, 4),
(6, '2020-06-18', 8, 4),
(8, '2020-06-15', 6, 3),
(8, '2020-06-18', 1, 3),
(8, '2020-06-22', 5, 3),
(9, '2020-06-15', 1, 4),
(9, '2020-06-18', 6, 5),
(9, '2020-06-22', 7, 5),
(10, '2020-06-15', 8, 5),
(10, '2020-06-18', 10, 4)
```

GO

3 Разработка основных операций реляционной алгебры

Для созданной базы данных необходимо создать и подробно описать не менее 15 запросов с использованием операций реляционной алгебры.

Примеры запросов для БД «Сессия».

1. Вывести полную информацию о студентах группы 945БО.

R = Студент WHERE №_группы = '945БО'

2. Вывести ФИО, должность и ученую степень преподавателей кафедры ЭВМ.

R = (Преподаватель WHERE Кафедра = 'ЭВМ') [ФИО, Должность, Ученая_степень]

3. Вывести список преподавателей, не имеющих ученую степень.
R = (Преподаватель WHERE Ученая_степень = NULL) [ФИО]
4. Вывести полную информацию о дисциплинах с указанием ФИО и должности преподавателя, который ведёт эту дисциплину.
R1 = Преподаватель RENAME Табельный_номер AS Преподаватель
R2 = (R1 JOIN Дисциплина) [Код_дисциплины, Наименование, Кол-во_часов, ФИО, Должность]
5. Вывести полную информацию обо всех преподавателях. Если преподаватель ведёт какие-то дисциплины, то укажите их наименования.
R1 = Дисциплина RENAME Преподаватель AS Табельный_номер
R2 = (Преподаватель LEFT JOIN Дисциплина) [Табельный_номер, ФИО, Кафедра, Должность, Ученая_степень, Стаж, Оклад, Наименование]
6. Вывести информацию о сдаче экзамена на оценки 4 и 5.
R1 = (Экзамен WHERE Оценка = 4)
R2 = (Экзамен WHERE Оценка = 5)
R3 = R1 UNION R2
7. Вывести ФИО и № группы студентов, которые не сдавали экзамены.
R1 = Студент [№_зачетки]
R2 = (Экзамен [Студент]) RENAME Студент AS №_зачетки
R3 = R1 EXCEPT R2
R4 = (R3 JOIN Студент) [ФИО, №_группы]
8. Вывести полную информацию о студентах, которые не сдавали Информатику.
R1 = Студент [№_зачетки]
R2 = Экзамен RENAME Дисциплина AS Код_дисциплины
R3 = R2 JOIN Дисциплина
R4 = (R3 WHERE Наименование = 'Информатика') [Студент]
R5 = R4 RENAME Студент AS №_зачетки
R6 = R1 EXCEPT R5
R7 = R6 JOIN Студент
9. Вывести ФИО студентов, которые сдавали Информатику, но не сдавали Программирование.
R1 = Экзамен RENAME Дисциплина AS Код_дисциплины
R2 = R1 JOIN Дисциплина
R3 = R2 WHERE Наименование = 'Информатика'
R4 = R2 WHERE Наименование = 'Программирование'
R5 = R3 EXCEPT R4
R6 = R5 RENAME Студент AS №_зачетки
R7 = (R6 JOIN Студент) [ФИО]
10. Вывести ФИО и № групп студентов, которые не сдавали ни Программирование, ни Информатику.

R1 = Студент [№_зачетки]
 R2 = Экзамен RENAME Дисциплина AS Код_дисциплины
 R3 = R2 JOIN Дисциплина
 R4 = (R3 WHERE Наименование = 'Программирование') [Студент]
 R5 = (R3 WHERE Наименование = 'Информатика') [Студент]
 R6 = R4 RENAME Студент AS №_зачетки
 R7 = R5 RENAME Студент AS №_зачетки
 R8 = (R1 EXCEPT R6) EXCEPT R7
 R9 = (R8 JOIN Студент) [ФИО, №_группы]
 11. Вывести ФИО студентов, которые сдали все дисциплины.
 R1 = Дисциплина [Код_дисциплины]
 R2 = Экзамен RENAME Дисциплина AS Код_дисциплины
 R3 = R2 [Студент, Код_дисциплины]
 R4 = R3 DIVIDE BY R1
 R5 = R4 RENAME Студент AS №_зачетки
 R6 = (R5 JOIN Студент) [ФИО]
 12. Вывести ФИО студентов, которые сдали все экзамены только на 5.
 R1 = (Экзамен WHERE Оценка = 5) [Студент]
 R2 = (Экзамен WHERE Оценка ≠ 5) [Студент]
 R3 = R1 EXCEPT R2
 R4 = R3 RENAME Студент AS №_зачетки
 R5 = (R4 JOIN Студент) [ФИО]
 13. Вывести информацию о сдаче студентами экзаменов в следующем виде:
 ФИО студента, Наименование дисциплины, ФИО преподавателя, Оценка, Дата
 R1 = Экзамен RENAME Студент AS №_зачетки, Дисциплина AS Код_дисциплины, Преподаватель AS Табельный_номер
 R2 = (R1 JOIN Студент) RENAME ФИО AS ФИО_студента
 R3 = (R2 JOIN Преподаватель) RENAME ФИО AS ФИО_преподавателя
 R4 = (R3 JOIN Дисциплина) RENAME Наименование AS Наименование_дисциплины
 R5 = R4 [ФИО_студента, Наименование_дисциплины, ФИО_преподавателя, Оценка, Дата]
 14. Вывести информацию о сдаче экзаменов, которые принимают преподаватели с должностью профессор, в следующем виде:
 ФИО студента, Наименование дисциплины, ФИО преподавателя, Должность, Оценка
 R1 = Экзамен RENAME Студент AS №_зачетки, Дисциплина AS Код_дисциплины, Преподаватель AS Табельный_номер
 R2 = (R1 JOIN Студент) RENAME ФИО AS ФИО_студента
 R3 = (R2 JOIN Преподаватель) RENAME ФИО AS ФИО_преподавателя
 R4 = R3 WHERE Должность = 'Профессор'

R5 = (R4 JOIN Дисциплина) RENAME Наименование AS
Наименование_дисциплины

R6 = R5 [ФИО_студента, Наименование_дисциплины,
ФИО_преподавателя, Должность, Оценка]

15. Вывести полную информацию о дисциплинах, по которым
студенты получали только удовлетворительные оценки.

R1 = (Экзамен WHERE Оценка = 3) [Дисциплина]

R2 = (Экзамен WHERE Оценка \neq 3) [Дисциплина]

R3 = R1 EXCEPT R2

R4 = R3 RENAME Дисциплина AS Код_дисциплины

R5 = R4 JOIN Дисциплина

4 Разработка запросов к БД для реализации информационных потребностей пользователя

Для созданной базы данных разработать на языке Transact-SQL следующие типы запросов:

1. Запросы с выбором строк по условию:
 - запрос с использованием операторов сравнения;
 - запрос с использованием логических операторов AND, OR и NOT;
 - запрос на использование комбинации логических операторов;
 - запрос на использование выражений над столбцами;
 - запрос с проверкой на принадлежность множеству;
 - запрос с проверкой на принадлежность диапазону значений;
 - запрос с проверкой на соответствие шаблону;
 - запрос с проверкой на неопределенное значение.
2. Запросы на теоретико-множественные реляционные операции:
 - запрос на объединение таблиц;
 - запрос на пересечение таблиц;
 - запрос на вычитание таблиц;
 - запрос с использованием декартового произведения таблиц;
3. Многотабличные запросы:
 - запрос с использованием соединения двух таблиц по равенству и условием отбора;
 - запрос с использованием соединения трех таблиц по равенству и условием отбора;
 - запрос с использованием соединения по четырем таблицам;
 - запрос с использованием левого внешнего соединения;
 - запрос на использование правого внешнего соединения;
 - запрос с использованием функции COUNT;
 - запрос с использованием функции SUM;
 - 3 запроса с использованием временных функций;
 - запрос с использованием группировки по одному столбцу;
 - запрос на использование группировки по нескольким столбцам;
 - запрос с использованием условия отбора групп HAVING;
 - запрос с использованием сортировки по столбцу;
4. Запросы на модификацию таблиц:
 - запрос на добавление новых данных в таблицу;
 - запрос на добавление новых данных по результатам выполненного запроса;
 - запрос на обновление существующих данных в таблице;
 - запрос на обновление существующих данных по результатам подзапроса во фразе WHERE;
 - запрос на удаление существующих данных;

- запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц.

- запрос на удаление существующих данных по результатам подзапроса во фразе WHERE;

- запрос на удаление существующих данных из одной таблицы на основе связанных с ней таблиц.

5. Многотабличные запросы с использованием подзапросов:

- 3 запроса с использованием операций сравнения;

- запрос с использованием операции ANY;

- запрос с использованием операции ALL;

- запрос с использованием операции IN;

- запрос с использованием операции EXISTS;

- запрос с использованием двух вложенных друг в друга подзапросов;

- запрос с использованием трех вложенных друг в друга подзапросов;

6. Запросы на изменение структуры существующей таблицы:

- запрос на добавление нового столбца к таблице;

- запрос на добавление нового ограничения к таблице;

- запрос на удаление нового столбца к таблице;

- запрос на удаление нового ограничения к таблице;

- запрос на изменение типа данных в таблице;

- запрос на добавление нового первичного ключа к таблице с сохранением целостности таблицы;