

## Содержание

Введение.....	3
1 Семантическое описание предметной области.....	4
2 Разработка базы данных.....	6
2.1 Разработка логической модели данных.....	6
2.2 Определение типов хранимых данных.....	10
2.3 Задание ограничений целостности данных.....	13
2.4 Построение схемы данных.....	15
2.5 Разработка сценария для создания БД и основных объектов структуры .....	16
2.6 Разработка сценария для заполнения таблиц БД.....	17
3 Разработка основных операций реляционной алгебры.....	21
4 Разработка запросов к БД для реализации информационных потребностей пользователя.....	23
Заключение.....	46
Список используемой литературы.....	47

## **Введение**

Современному человеку необходимо узнавать, изменять и обрабатывать взаимосвязанную информацию. В 21 веке количество информации выросло в разы, поэтому хранение информации доверили базам данных.

База данных (БД) — это организованная структура, предназначенная для хранения, изменения и обработки взаимосвязанной информации, преимущественно больших объемов.

Для изменения и управления данными существуют системы управления базами данных (СУБД). Это комплекс программно-языковых средств, позволяющих создать БД и управлять данными.

Целью выполнения данного курсового проекта является систематизация, закрепление и углубление теоретических знаний и практических навыков проектирования баз данных.

## **1 Семантическое описание предметной области**

Требуется разработать информационную систему для учета и хранения информации о продаже абонементов в бассейн. Допускаем, что возможность использования бассейна дается только после приобретения определенного абонемента, следовательно, база данных должна хранить информацию об абонементе, каждый из которых характеризуется следующими параметрами:

- уникальный код или номер;
- название;
- описание;
- стоимость;
- срок действия.

Чтобы получить данный абонемент необходимо заключить договор между клиентом и организацией, в которой находится бассейн. Следовательно, с каждым клиентом заключается договор, в который заносятся следующие сведения:

- номер договора;
- код клиента;
- код абонемента;
- дата и время покупки абонемента.

Чтобы идентифицировать клиента необходимо иметь следующую информацию о нем:

- код клиента;
- фамилия, имя, отчество;
- дата рождения;
- адрес;
- телефон.

В случае, если клиент не умеет плавать, должна иметься возможность получения индивидуальной тренировки с инструктором. На каждого человека не возможно дать личного тренера, следовательно необходимо хранить информацию о том, в какое время и с каким клиентом у тренера будет занятие. Поэтому персональная тренировка характеризуется:

- кодом клиента;
- датой и временем тренировки;
- стоимостью;
- фамилией, именем и отчеством инструктора, который проводит тренировку.

Таким образом, база данных «Бассейн» будет иметь четыре отношения, которые будут хранить необходимую информацию:

1. Тип абонеента (Код, Название, Описание, Стоимость, Срок действия);
2. Клиенты (Код клиента, ФИО, Дата рождения, Адрес, Телефон);
3. Договор (Номер договора, код клиента, код абонеента, Дата и время покупки);
4. Персональные тренировки (Код клиента, Дата и время тренировки, Стоимость, ФИО инструктора).

Абонемент на плавание выдается на основе договора, заключенного между клиентом и организацией. Договор заключается с конкретным клиентом. Персональная тренировка может пройти, если клиенту она понадобится и он договорится с тренером.

## 2 Разработка базы данных

### 2.1 Разработка логической модели данных

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

Тип абонемента, Клиенты, Договор, Персональные тренировки.

Представим список сущностей предметной области (таблица 1).

Таблица 1 — Список сущностей

№	Название	Назначение
1	Тип абонемента	Перечень абонементов с их описанием, которые могут приобрести клиенты
2	Клиенты	Основная информация о клиенте
3	Договор	Перечень договоров, заключенных между клиентом и организацией
4	Персональные тренировки	Информация о тренере, стоимости и дате проведения индивидуальной тренировки с клиентом

Для каждой сущности приведем описание ее атрибутов в виде таблицы. Выделим ключевые и не ключевые атрибуты. Результаты приведены в таблицах 2-5.

Следует отметить, что для сущностей Тип абонемента, Клиенты и Договор добавлены атрибуты с кодом. Эти атрибуты называются суррогатными, их предназначение — служить первичным ключом. При добавлении новой записи в таблицу автоматически записывается уникальное для этой таблицы числовое значение — так называемый «автоинкремент» (в Microsoft SQL Server – свойство IDENTITY) или ID. Можно указать начальное значение и значение приращения, которое прибавляется к предыдущему значению на каждом шаге.

Таблица 2 — Список атрибутов таблицы «Тип абонемента»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому абонементу. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
	Название	
	Описание	
	Стоимость	
	Срок действия	

Таблица 3 — Список атрибутов таблицы «Клиенты»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Код клиента	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому клиенту. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
	ФИО	
	Дата рождения	
	Адрес	
	Телефон	

Таблица 4 — Список атрибутов таблицы «Договор»

Ключевое поле	Название	Назначение
ПК (первичный ключ)	Номер договора	Ключевое поле. Представляет собой первичный ключ. Это уникальное значение, соответствующее каждому договору. Значения автоматически генерируется СУБД при вставке новой записи в таблицу.
ВК (внешний ключ)	Код клиента	С помощью данного внешнего ключа можно определить, какой клиент заключил данный договор
ВК (внешний ключ)	Код абонемента	С помощью данного внешнего ключа можно определить, на какой абонемент был заключен договор
	Дата и время покупки	

Таблица 5 — Список атрибутов таблицы «Персональные тренировки»

Ключевое поле	Название	Назначение
ПК(первичный ключ) ВК (внешний ключ)	Код клиента	С помощью данного внешнего ключа можно определить, какой клиент записался на тренировку
ПК(первичный ключ)	Дата и время тренировки	Составной первичный ключ, Код клиента и Дата и время тренировки. Это уникальное сочетание значений, соответствующее каждой тренировке.
	Стоимость	
	ФИО инструктора	

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 приведем обоснование типов связей (1:1, 1:N, N:N).

Результат представим 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	Тип абонемента — Договор	1:N	Договор может заключаться на разные типы абонемента, однако на один договор может приходиться один абонемент
2	Клиенты — Договор	1:N	Один клиент может заключать несколько договоров, однако один конкретный договор может быть заключен только с одним клиентом
3	Клиенты — Тренировки	1:N	Один клиент может тренироваться N раз, однако на конкретную персональную тренировку может прийти только один клиент





Рисунок 1 — Логическая модель предметной области «Бассейн»

## 2.2 Определение типов хранимых данных

Чтобы определить типы хранимых данных, необходимо проанализировать, что будет представлять из себя каждый атрибут какого-либо отношения.

Рассмотрим отношение «Тип абонемента». В нем используются 5 атрибутов, такие как: «Код абонемента», «Название», «Описание», «Стоимость», «Срок действия».

«Код абонемента» будет хранить в себе некий набор цифр, поэтому будет использоваться тип данных `int`.

«Название абонемента» и «Описание» будут состоять из некого набора букв — слова или слов, поэтому эти атрибуты будут использовать тип данных `nvarchar`.

Атрибут «Стоимость» будет хранить в себе какую-то сумму денег. Для этого лучше всего подходит тип данных `money`. Однако, нет необходимости хранить огромные суммы, следовательно будем использовать `smallmoney`.

Атрибуту «Срок действия» будет хранить какую-то дату, поэтому лучше всего подойдет тип данных `date`.

Проанализировав отношение «Абонементы», выбрали для каждого атрибута тип данных. Представим это в виде таблицы (таблица 7).

Таблица 7 — Список атрибутов таблицы «Тип абонемента» и типы данных

Тип абонемента	
Название	Тип данных
Код	int, нач. значение = 1, приращение = 1
Название	nvarchar(100)
Описание	nvarchar(100)
Стоимость	smallmoney
Срок действия	date

Таким же образом, проанализировав остальные отношения, получим еще 3 таблицы (таблицы 8-10), в которых как и в таблице 7 запишем типы данных.

Таблица 8 — Список атрибутов таблицы «Клиенты» и типы данных

Клиенты	
Название	Тип данных
Код клиента	int, нач. значение = 1, приращение = 1
ФИО	nvarchar(100)
Дата рождения	date
Адрес	nvarchar(100)
Телефон	nvarchar(100)

Таблица 9 — Список атрибутов таблицы «Договор» и типы данных

Договор	
Название	Тип данных
Номер договора	int
Код клиента	int
Код абонента	int
Дата и время покупки	datetime

Таблица 10 — Список атрибутов таблицы «Персональные тренировки»  
и типы данных

Персональные тренировки	
Название	Тип данных
Код клиента	int
Дата и время тренировки	datetime
Стоимость	smallmoney
ФИО инструктора	nvarchar(100)

В таблице 8 в атрибуте «Телефон» тип данных указан `nvarchar`, хотя, казалось бы, номер телефона состоит из цифр. Это сделано для того, чтобы из-

бежать ошибок при попытке ввода номеров телефонов вида: +7 (4912) 50-40-20.

### 2.3 Задание ограничений целостности данных

Чтобы задать ограничение целостности данных, необходимо рассмотреть, какие ограничения есть.

Существуют следующие типы ограничений:

1. Ограничение **NOT NULL** дает возможность запретить использование в указанном столбце NULL-значений. Такое ограничение может быть указано только как ограничение на столбец.
2. Ограничение **UNIQUE**. Ограничение разрешает использовать только уникальные значения для указанных столбцов.
3. Ограничение первичных ключей **PRIMARY KEY (PK)**. Это специальный случай комбинирования ограничений **UNIQUE** и **NOT NULL**. Простой первичный ключ задается как ограничение на конкретный столбец.
4. Ограничение внешнего ключа **FOREIGN KEY(FK)**. Определяется в дочерней таблице и задает ее связь с родительской таблицей
5. Ограничение на проверку **CHECK**. Задает множество возможных значений атрибута. Оно записывается как ограничение на столбец или таблицу. Для одного столбца может быть задано несколько ограничений.
6. Ограничение **DEFAULT**. Позволяет задать значение по умолчанию. Ограничение задается только как ограничение на столбец.
7. При определении столбца в операторе **CREATE TABLE** можно определить столбец-счетчик, значения которого будут автоматически генерироваться сервером. Для этого нужно указать свойство **IDENTITY(ID)** (<начальное значение >, <шаг>). Свойство **IDENTITY** может быть указано только для одного столбца в таблице. Обычно оно указывается для атрибута, выбранного в качестве первичного ключа.

Используя таблицы 7-10 рассмотрим, к каким атрибутам можно применить ограничения. Запишем их в таблицах 11-14.

Таблица 11 — Ограничения таблицы «Тип абонемента»

Тип абонемента	
Название	Ограничение
Код	PK, ID
Название	UNIQUE, CHECK LIKE('[А-Я]%%'), NOT NULL
Описание	CHECK LIKE('[А-Я]%%'), NOT NULL
Стоимость	NOT NULL, CHECK >0
Срок действия	NOT NULL

Таблица 12 — Ограничения таблицы «Клиенты»

Клиенты	
Название	Ограничение
Код клиента	PK, ID
ФИО	NOT NULL, CHECK LIKE('[А-Я]%%')
Дата рождения	NOT NULL, CHECK (YEAR<2019)
Адрес	
Телефон	NOT NULL, CHECK LIKE('+7 ([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')

Таблица 13 — Ограничения таблицы «Договор»

Договор	
Название	Ограничение
Номер договора	PK, ID
Код клиента	FK(Клиенты), NOT NULL
Код абонемента	FK(Абонементы), NOT NULL
Дата и время покупки	NOT NULL

Таблица 14 — Ограничения таблицы «Персональные тренировки»

Персональные тренировки	
Название	Ограничение
Код клиента	FK(Клиенты), PK
Дата и время тренировки	UNIQUE(ФИО инструктор), NOT NULL, PK
Стоимость	NOT NULL, CHECK>0
ФИО инструктора	UNIQUE(Дата время тренировки), NOT NULL

## 2.4 Построение схемы данных

Схема БД представляет собой набор заголовков таблиц. То есть на схеме БД указываются имена таблиц (сущностей), их атрибуты и связи между таблицами.

На основе информации, полученной в пунктах 2.1-2.3 построим схему данных будущей БД.

Полученная схема представлена на рисунке 2.

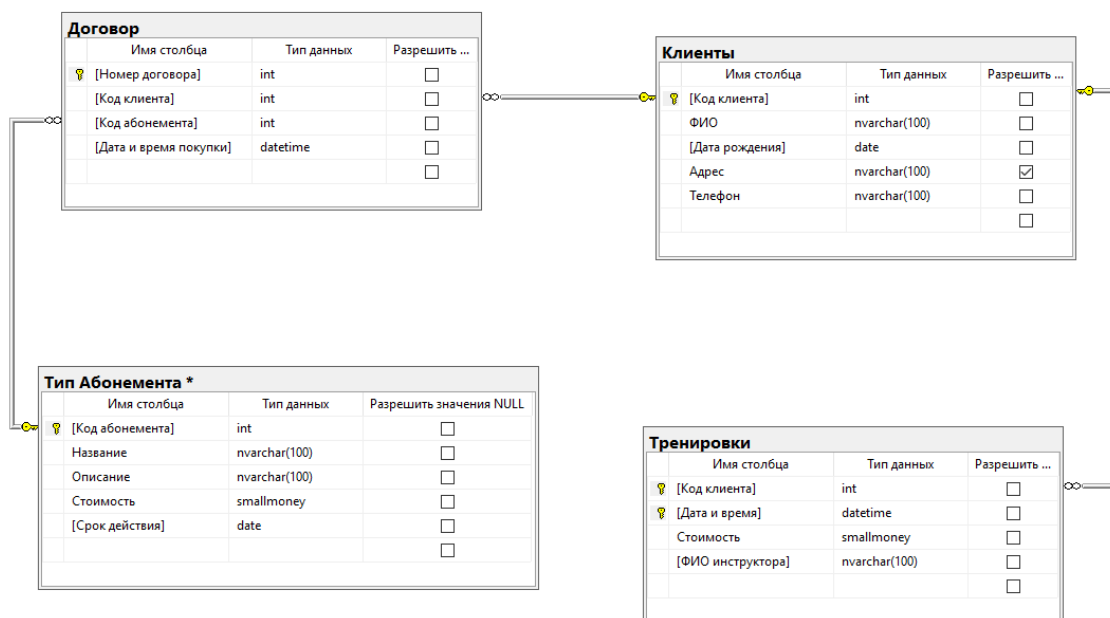


Рисунок 2 — Схема данных

## 2.5 Разработка сценария для создания БД и основных объектов структуры

После разработки сценария для создания БД получили код, представленный ниже.

```
use master
drop database SwimmingPool_945
--Создаем БД--
USE MASTER
CREATE DATABASE SwimmingPool_945
ON
(NAME='Swiming_Pool_945',
FILENAME='H:\university\3 семестр\Базы Данных\KP\Итог\Swimming_Pool_945.mdf',
SIZE=2,
MAXSIZE=10,
FILEGROWTH=2)
LOG ON
(NAME='SP_945_Log',
FILENAME='H:\university\3 семестр\Базы Данных\KP\Итог\Swimming_Pool_945_Log.mdf',
SIZE=2,
MAXSIZE=10,
FILEGROWTH=2)
GO
--Создаем таблицу Тип Абонемента--
use SwimmingPool_945
CREATE TABLE [Тип Абонемента]
([Код абонемента] int identity (1,1) NOT NULL,
Название nvarchar(100) NOT NULL UNIQUE check (Название like ('[А-Я][а-я]%'')),
Описание nvarchar(100) NOT NULL check (Описание like ('[А-Я][а-я]%'')),
Стоимость smallmoney NOT NULL CHECK(Стоимость>0),
[Срок действия] date NOT NULL)
GO
--Устанавливаем ограничение PK таблицы Тип абонемента--
ALTER TABLE [Тип Абонемента]
ADD CONSTRAINT PK_ТипАбонемента primary key ([Код абонемента])
GO
--Создаем таблицу Клиенты--
CREATE TABLE Клиенты
([Код клиента] int identity (1,1) NOT NULL,
ФИО nvarchar(100) NOT NULL check (ФИО like ('[А-Я][а-я]%'')),
[Дата рождения] date check (year([Дата рождения])<2019) NOT NULL,
Адрес nvarchar (100) ,
Телефон nvarchar (100) NOT NULL check ([Телефон] like '+7 ([0-9][0-9][0-9]) [0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]-[0-9][0-9]'))
GO
--Устанавливаем ограничение PK таблицы Клиенты--
ALTER TABLE Клиенты
ADD CONSTRAINT PK_Клиенты primary key ([Код клиента])
go
--Создаем таблицу Договор--
CREATE TABLE Договор
([Номер договора] int IDENTITY (1,1) NOT NULL,
[Код клиента]int NOT NULL,
[Код абонемента] int NOT NULL,
[Дата и время покупки] datetime NOT NULL)
```

```

GO
--Устанавливаем ограничение PK таблицы Договор--
ALTER TABLE Договор
ADD CONSTRAINT PK_Договор primary key ([Номер договора])
go
--Устанавливаем ограничение FK таблицы договор--
ALTER TABLE Договор
add CONSTRAINT ДОГОВОР_ТИПАБОНЕМЕНТА_FK
FOREIGN KEY ([Код абонемента]) REFERENCES [Тип Абонемента]([Код абонемента])
ON DELETE CASCADE
ON UPDATE CASCADE
go
--Устанавливаем ограничение FK таблицы договор--
ALTER TABLE Договор
add CONSTRAINT ДОГОВОР_КЛИЕНТЫ_FK
FOREIGN KEY ([Код клиента]) REFERENCES [Клиенты]([Код клиента])
ON DELETE CASCADE
ON UPDATE CASCADE
go
--Создаем таблицу Тренировки--
CREATE TABLE Тренировки
([Код клиента]int NOT NULL,
[Дата и время] datetime NOT NULL,
Стоимость smallmoney NOT NULL Check(Стоимость>0),
[ФИО инструктора] nvarchar(100) NOT NULL)
go
--Устанавливаем ограничение PK таблицы Тренировки--
alter table Тренировки
add constraint PK_Тренировки
primary key ([Код клиента],[Дата и время])
go
--Устанавливаем ограничение FK таблицы Тренировки--
alter table Тренировки
add CONSTRAINT ТРЕНИРОВКИ_КЛИЕНТЫ_FK
FOREIGN KEY ([Код клиента]) REFERENCES [Клиенты]([Код клиента])
ON DELETE CASCADE
ON UPDATE CASCADE
go
--Устанавливаем ограничение UNIQUE таблицы Тренировки--
alter table Тренировки
add CONSTRAINT ДАТА_ФИО_UNIQUE
UNIQUE ([Дата и время], [ФИО инструктора])
GO

```

## 2.6 Разработка сценария для заполнения таблиц БД

После разработки сценария для заполнения таблиц БД получили код, представленный ниже.

```

use master
use SwimmingPool
go
--Заполнение таблицы Клиенты--
DBCC CHECKIDENT('Клиенты', RESEED, 1) --сброс счетчика
go

```



```

insert into Клиенты
values
( 'Иванов Иван Иванович', '1975-10-25', 'Пушкина 8', '+7 (915) 558-65-69'),
( 'Михайлов Дмитрий Владимирович', '1980-08-14', 'Грибоедова 19', '+7 (930) 667-56-58'),
( 'Леонов Михаил Максимович', '2006-02-5', 'Васильевская 8', '+7 (800) 535-35-35'),
( 'Чуваев Артем Александрович', '2000-09-23', 'Тимуровцев 15', '+7 (910) 999-53-53'),
( 'Дмитриев Дмитрий Иванович', '1975-06-22', 'Новоселов 35', '+7 (992) 321-22-33'),
( 'Смольский Иван Олегович', '2004-10-10', 'Московский 2', '+7 (920) 258-45-65'),
( 'Егорова Елена Васильевна', '2008-12-22', 'Горького 16', '+7 (910) 456-69-62'),
( 'Олегова Алина Евгеньевна', '1990-03-14', 'Гоголя 23', '+7 (932) 999-44-55'),
( 'Вольская Ирина Чернова', '1999-03-25', 'Пушкина 25', '+7 (920) 356-44-25'),
( 'Одинцова Ольга Владимировна', '2009-03-8', 'Горького 63', '+7 (920) 321-22-55')
go
insert into Клиенты(ФИО, [Дата рождения], Телефон)
values
('Житков Ильдар Викторович', '1980-05-02', '+7 (920) 888-77-66')
go
--Заполнение таблицы Тип абонемента--
DBCC CHECKIDENT('Тип Абонемента', RESEED, 1) --сброс счетчика
go
insert into [Тип Абонемента]
values
('Разовый детский', 'Одно посещение ребенка', 200, '2020-12-31'),
('Разовый взрослый', 'Одно посещение взрослого', 300, '2020-12-31'),
('Разовый студенческий', 'Одно посещение студента', 100, '2020-12-31'),
('Пятиразовый', 'Пять посещений', 800, '2020-12-31'),
('Месячный взрослый', 'Месяц без ограничений', 1500, '2021-12-31'),
('Месячный студенческий', 'Месяц без ограничений студентов', 1000, '2021-12-31'),
('Пятиразовый студенческий', 'Пять посещений студента', 500, '2021-12-31'),
('Трехмесячный', 'Пять месяцев без ограничений', 3000, '2021-12-31'),
('Полугодовой', 'Полгода без ограничений', 5000, '2022-12-31'),
('Трехмесячный студенческий', 'Три месяца студента без ограничений', 1900, '2021-12-31')
go
--Заполнение таблицы Тренировки--
insert into Тренировки
values
(1, '2020-11-06T12:00:00', 300, 'Попов Юрий Александрович'),
(2, '2020-11-02T13:00:00', 300, 'Попов Юрий Александрович'),
(4, '2020-11-13T16:00:00', 100, 'Николаев Артем Викторович'),
(3, '2020-11-09T18:00:00', 200, 'Филлипов Евгений Геннадиевич'),
(6, '2020-11-08T12:00:00', 300, 'Филлипов Евгений Геннадиевич'),

```

```

(7, '2020-11-13T11:00:00', 200, 'Попов Юрий Александрович'),
(5, '2020-11-26T09:00:00', 300, 'Попов Юрий Александрович'),
(8, '2020-11-20T20:00:00', 300, 'Филлипов Евгений Геннадиевич'),
(10, '2020-11-19T14:00:00', 200, 'Попов Юрий Александрович'),
(9, '2020-11-15T12:00:00', 100, 'Николаев Артем Викторович'),
(7, '2020-11-16T20:00:00', 200, 'Попов Юрий Александрович'),
(1, '2020-11-26T15:00:00', 300, 'Попов Юрий Александрович'),
(10, '2020-11-29T13:00:00', 200, 'Попов Юрий Александрович'),
(2, '2020-11-20T21:00:00', 300, 'Попов Юрий Александрович'),
(4, '2020-11-23T10:00:00', 100, 'Николаев Артем Викторович'),
(3, '2020-11-16T13:00:00', 200, 'Филлипов Евгений Геннадиевич'),
(5, '2020-11-26T18:00:00', 300, 'Попов Юрий Александрович'),
(6, '2020-11-30T19:00:00', 200, 'Филлипов Евгений Геннадиевич'),
(8, '2020-11-12T19:00:00', 300, 'Филлипов Евгений Геннадиевич'),
(1, '2020-11-11T18:00:00', 300, 'Попов Юрий Александрович'),
(4, '2020-11-16T11:00:00', 100, 'Николаев Артем Викторович'),
(8, '2020-11-16T12:00:00', 300, 'Филлипов Евгений Геннадиевич')

```

go

--Заполнение таблицы Договор--

DBCC CHECKIDENT('Договор', RESEED, 1) --сброс счетчика

go

insert into Договор

values

```

(1,2,'2020-01-20T12:31:00'),
(2,4,'2020-10-30T13:18:00'),
(4,3,'2020-10-22T10:35:00'),
(5,2,'2020-10-23T09:45:00'),
(6,1,'2020-10-11T14:58:00'),
(3,1,'2020-10-18T19:23:00'),
(10,1,'2020-10-23T20:20:00'),
(9,3,'2020-10-24T20:00:00'),
(7,1,'2020-10-13T19:05:00'),
(8,2,'2020-10-09T18:13:00'),
(10,1,'2020-10-27T21:14:00'),
(9,3,'2020-10-11T20:54:00'),
(8,2,'2020-10-30T11:11:00'),
(6,1,'2020-10-21T10:21:00'),
(7,1,'2020-10-14T21:17:00'),
(2,1,'2020-10-04T22:36:00'),
(1,2,'2020-10-10T13:32:00'),
(2,2,'2020-10-10T13:32:00'),

```

```
(3,1,'2020-10-10T13:32:00'),  
(1,2,'2020-10-20T12:31:00'),  
(1,2,'2020-10-20T12:32:00'),  
(1,2,'2020-10-20T12:33:00')  
go
```

### 3 Разработка основных операций реляционной алгебры

1. Вывести ФИО и дату рождения клиентов.

R=Клиенты[ФИО],[Дата рождения]

2. Вывести ФИО и номер клиентов, у которых отсутствует адрес.

R=(Клиенты WHERE Адрес IS NULL) [ФИО][Телефон]

3. Вывести названия абонементов у клиентов с кодами 1-5.

R1= (Договор JOIN Тип абонемента)

R2=(R1 WHERE Код клиента BETWEEN 1 and 5)[Код клиента][Название]

4. Вывести несовершеннолетних клиентов.

R=Клиенты WHERE YEAR(Дата рождения)>2002

5. Вывести клиентов тренера Филипова

R1=Тренировки JOIN Клиенты

R2= (R1 WHERE ФИО инструктора=Филипов Евгений Геннадиевич )  
[ФИО]

6. Клиенты со студенческим абонементом.

R1=(Тип абонемента JOIN Договор JOIN Клиенты)

R2= (R1 WHERE Название LIKE(%студенческий))[ФИО]

7. Вывести клиентов, у которых нет персональных тренировок.

R1=(Клиенты JOIN Тренировки)

R2=R1[Код клиента][ФИО][Дата рождения][Адрес][Телефон]

R3=Клиенты EXCEPT R2

8. Вывести клиентов, у которых тренер Попов.

R1=(Тренировки WHERE ФИО инструктора=Попов Юрий Александрович)[Код клиента]

R2=(Клиенты DIVIDE BY R1)[ФИО]

9. Вывести код клиентов, у которых месячный взрослый абонемент

R1=Договор JOIN Тип абонемента

R2=(R1 WHERE Название="Месячный взрослый")[Код клиента]

10. Вывести совершеннолетних клиентов

R1=Клиенты WHERE Дата рождения>=2002-01-01

11. Вывести клиентов, у которых номер договора начинается с 2

R1=Клиенты JOIN Договор

R2=(R1 WHERE Номер договора like ('2%'))[ФИО]

12. Вывести ФИО клиентов у которых стоимость тренировки >100

R1=Клиенты JOIN Тренировки

R2=(R1 WHERE Стоимость>100) [ФИО]

13. Вывести информацию о клиентах, у которых указан адрес

R=Клиенты WHERE Адрес IS NOT NULL

14. Вывести всю информацию об абонементе, начинающегося с 3

R=Тип абонемента WHERE Код абонемента like ('3%')

15. ФИО клиентов, у которых разовое занятие

R1=Клиенты JOIN Договор JOIN Тип Абонемента

R2= (R1 WHERE Название LIKE ('Разовый%'))[ФИО]

16. Вывести клиентов Попова и Филипова

R1=(Тренировки WHERE ФИО инструктора=Попов Юрий Александрович)[Код клиента]

R2=(Клиенты DIVIDE BY R1)[ФИО]

R3=Тренировки JOIN Клиенты

R4= (R1 WHERE ФИО инструктора=Филипов Евгений Геннадиевич )  
[ФИО]

R5=R2 UNION R3

17. Вывести коды клиентов у которых есть тренировки

R1=Клиенты [Код]

R2=Тренировки[Код клиента]

R3=R1 INTERSECT R2

## 4 Разработка запросов к БД для реализации информационных потребностей пользователя

Для созданной базы данных разработаем следующие типы запросов:

1. Запросы с выбором строк по условию:

- запрос с использованием операторов сравнения;

<pre>--Вывести всю информацию об абонементх, с номером &gt;3 select * from [Тип Абонемента] where [Код абонемента]&gt;3</pre>					
	Код абонемента	Название	Описание	Стоимость	Срок действия
1	4	Пятиразовый	Пять посещений	800,00	2020-12-31
2	5	Месячный взрослый	Месяц без ограничений	1500,00	2021-12-31
3	6	Месячный студенческий	Месяц без ограничений студентов	1000,00	2021-12-31
4	7	Пятиразовый студенческий	Пять посещений студента	500,00	2021-12-31
5	8	Трехмесячный	Пять месяцев без ограничений	3000,00	2021-12-31
6	9	Полугодовой	Полгода без ограничений	5000,00	2022-12-31
7	10	Трехмесячный студенческий	Три месяца студента без ограничений	1900,00	2021-12-31

Рисунок 4.1.1 — Запрос с использованием операторов сравнения

- запрос с использованием логических операторов AND, OR и NOT;

<pre>--вывести всю информацию об абонементх, в которых нет слова Разовый select * from [Тип Абонемента] where Название not like ('Разовый%')</pre>					
	Код абонемента	Название	Описание	Стоимость	Срок действия
1	4	Пятиразовый	Пять посещений	800,00	2020-12-31
2	5	Месячный взрослый	Месяц без ограничений	1500,00	2021-12-31
3	6	Месячный студенческий	Месяц без ограничений студентов	1000,00	2021-12-31
4	7	Пятиразовый студенческий	Пять посещений студента	500,00	2021-12-31
5	8	Трехмесячный	Пять месяцев без ограничений	3000,00	2021-12-31
6	9	Полугодовой	Полгода без ограничений	5000,00	2022-12-31
7	10	Трехмесячный студенческий	Три месяца студента без ограничений	1900,00	2021-12-31

Рисунок 4.1.2 — Запрос с использованием логических операторов AND, OR и NOT

- запрос на использование комбинации логических операторов;

<pre>--вывести всю информацию о клиентах, с кодом оператора 910 или не 920 select * from Клиенты where Телефон like ('%910%') or not Телефон like ('%920%')</pre>					
	Код клиента	ФИО	Дата рождения	Адрес	Телефон
1	1	Иванов Иван Иванович	1975-10-25	Пушкина 8	+7 (915) 558-65-69
2	2	Михайлов Дмитрий Владимирович	1980-08-14	Грибоедова 19	+7 (930) 667-56-58
3	3	Леонов Михаил Максимович	2006-02-05	Васильевская 8	+7 (800) 535-35-35
4	4	Чуваев Артем Александрович	2000-09-23	Тимуровцев 15	+7 (910) 999-53-53
5	5	Дмитриев Дмитрий Иванович	1975-06-22	Новоселов 35	+7 (992) 321-22-33
6	7	Егорова Елена Васильевна	2008-12-22	Горького 16	+7 (910) 456-69-62
7	8	Олегова Алина Евгеньевна	1990-03-14	Гоголя 23	+7 (932) 999-44-55

Рисунок 4.1.3 — Запрос на использование комбинации логических операторов

- запрос на использование выражений над столбцами;

<pre>--вывести студенческие абонементы в руб. и \$ select [Код абонемент],Стоимость as [Стоимость руб], Стоимость/73.19 as [Стоимость \$] from [Тип Абонемент]</pre>			
	Код абонемент	Стоимость руб	Стоимость \$
1	1	200,00	2.732613745
2	2	300,00	4.098920617
3	3	100,00	1.366306872
4	4	800,00	10.930454980
5	5	1500,00	20.494603087
6	6	1000,00	13.663068725
7	7	500,00	6.831534362
8	8	3000,00	40.989206175
9	9	5000,00	68.315343626
10	10	1900,00	25.959830577

Рисунок 4.1.4 — Запрос на использование выражений над столбцами

- запрос с проверкой на принадлежность множеству;

<pre>--вывести всю информацию о тренировках, стоимостью 300 select * from Тренировки where Стоимость in (300)</pre>	
---	--

	Код клиента	Дата и время	Стоимость	ФИО инструктора
1	1	2020-11-06 12:00:00.000	300,00	Попов Юрий Александрович
2	2	2020-11-02 13:00:00.000	300,00	Попов Юрий Александрович
3	6	2020-11-08 12:00:00.000	300,00	Филлипов Евгений Геннадиевич
4	5	2020-11-26 09:00:00.000	300,00	Попов Юрий Александрович
5	8	2020-11-20 20:00:00.000	300,00	Филлипов Евгений Геннадиевич
6	1	2020-11-26 15:00:00.000	300,00	Попов Юрий Александрович
7	2	2020-11-20 21:00:00.000	300,00	Попов Юрий Александрович
8	5	2020-11-26 18:00:00.000	300,00	Попов Юрий Александрович
9	8	2020-11-12 19:00:00.000	300,00	Филлипов Евгений Геннадиевич
10	1	2020-11-11 18:00:00.000	300,00	Попов Юрий Александрович
11	8	2020-11-16 12:00:00.000	300,00	Филлипов Евгений Геннадиевич

Рисунок 4.1.5 — Запрос с проверкой на принадлежность множеству

- запрос с проверкой на принадлежность диапазону значений;

```
--вывести названия абонементов, цена которых в диапазоне 500 и 1000
select Название
from [Тип Абонемента]
where Стоимость between 500 and 1000
```

	Название
1	Пятиразовый
2	Месячный студенческий
3	Пятиразовый студенческий

Рисунок 4.1.6 — Запрос с проверкой на принадлежность диапазону значений

- запрос с проверкой на соответствие шаблону;

```
--вывести дату и время тренировок Попова Ю.А.
select [ФИО инструктора], [Дата и время]
from Тренировки
where [ФИО инструктора] like ('Попов%')
```



	ФИО инструктора	Дата и время
1	Попов Юрий Александрович	2020-11-02 13:00:00.000
2	Попов Юрий Александрович	2020-11-06 12:00:00.000
3	Попов Юрий Александрович	2020-11-11 18:00:00.000
4	Попов Юрий Александрович	2020-11-13 11:00:00.000
5	Попов Юрий Александрович	2020-11-16 20:00:00.000
6	Попов Юрий Александрович	2020-11-19 14:00:00.000
7	Попов Юрий Александрович	2020-11-20 21:00:00.000
8	Попов Юрий Александрович	2020-11-26 09:00:00.000
9	Попов Юрий Александрович	2020-11-26 15:00:00.000
10	Попов Юрий Александрович	2020-11-26 18:00:00.000
11	Попов Юрий Александрович	2020-11-29 13:00:00.000

Рисунок 4.1.7 — Запрос с проверкой на соответствие шаблону

- запрос с проверкой на неопределенное значение.

```
--вывести ФИО и номер клиента, у которого не указан адрес
select ФИО, Телефон
from Клиенты
where Адрес is null
```

	ФИО	Телефон
1	Житков Ильдар Викторович	+7 (920) 888-77-66

Рисунок 4.1.8 — Запрос с проверкой на неопределенное значение

## 2. Запросы на теоретико-множественные реляционные операции:

- запрос на объединение таблиц;

```
--вывести ФИО клиентов и названия их абонементов
select distinct ФИО, Название as Абонемент
from [Тип Абонемента] join Договор on [Тип Абонемента].[Код абонемента]=Договор.
[Код абонемента] join Клиенты on Договор.[Код клиента]=Клиенты.[Код клиента]
```

	ФИО	Абонемент
1	Вольская Ирина Чернова	Разовый студенческий
2	Дмитриев Дмитрий Иванович	Разовый взрослый
3	Егорова Елена Васильевна	Разовый детский
4	Иванов Иван Иванович	Разовый взрослый
5	Леонов Михаил Максимович	Разовый детский
6	Михайлов Дмитрий Владимирович	Пятиразовый
7	Михайлов Дмитрий Владимирович	Разовый взрослый
8	Михайлов Дмитрий Владимирович	Разовый детский
9	Одинцова Ольга Владимировна	Разовый детский
10	Олегова Алина Евгеньевна	Разовый взрослый
11	Смольский Иван Олегович	Разовый детский
12	Чуваев Артем Александрович	Разовый студенческий

Рисунок 4.2.1 — Запрос на объединение таблиц

- запрос на пересечение таблиц;

```
--вывести коды клиентов, у которых студенческий абонемент и есть персональные
тренировки
select [Код клиента]
from Тренировки
intersect
select [Код клиента]
from Договор
where [Код абонемента] in (select [Код абонемента] from [Тип Абонемента] where
Название like ('% студенческий'))
```

	Код клиента
1	4
2	9

Рисунок 4.2.2 — Запрос на пересечение таблиц

- запрос на вычитание таблиц;

```
--вывести коды клиентов, у которых нет персональных тренировок
select [Код клиента]
from Клиенты
except
select [Код клиента]
from Тренировки
```

	Код клиента
1	11

Рисунок 4.2.3 — Запрос на вычитание таблиц

- запрос с использованием декартового произведения таблиц;

```
--вывести всю информацию о клиентах и тренировках
select distinct *
from Тренировки cross join Клиенты
where Тренировки.[Код клиента]=Клиенты.[Код клиента]
```

	Код клиента	Дата и время	Стоимость	ФИО инструктора	Код клиента	ФИО	Дата рождения	Адрес	Телефон
1	1	2020-11-06 12:00:00.000	300,00	Попов Юрий Александрович	1	Иванов Иван Иванович	1975-10-25	Пушкина 8	+7 (915) 558-65-69
2	2	2020-11-02 13:00:00.000	300,00	Попов Юрий Александрович	2	Михайлов Дмитрий Владимирович	1980-08-14	Грибоедова 19	+7 (930) 667-56-58
3	4	2020-11-13 16:00:00.000	100,00	Николаев Артем Викторович	4	Чуваев Артем Александрович	2000-09-23	Тимуровцев 15	+7 (910) 999-53-53
4	3	2020-11-09 18:00:00.000	200,00	Филлипов Евгений Геннадиевич	3	Леонов Михаил Максимович	2006-02-05	Васильевская 8	+7 (800) 535-35-35
5	6	2020-11-08 12:00:00.000	300,00	Филлипов Евгений Геннадиевич	6	Смольский Иван Олегович	2004-10-10	Московский 2	+7 (920) 258-45-65
6	7	2020-11-13 11:00:00.000	200,00	Попов Юрий Александрович	7	Егорова Елена Васильевна	2008-12-22	Горького 16	+7 (910) 456-69-62
7	5	2020-11-26 09:00:00.000	300,00	Попов Юрий Александрович	5	Дмитриев Дмитрий Иванович	1975-06-22	Новоселов 35	+7 (992) 321-22-33
8	8	2020-11-20 20:00:00.000	300,00	Филлипов Евгений Геннадиевич	8	Олегова Алина Евгеньевна	1990-03-14	Гоголя 23	+7 (932) 999-44-55
9	10	2020-11-19 14:00:00.000	200,00	Попов Юрий Александрович	10	Одинцова Ольга Владимировна	2009-03-08	Горького 63	+7 (920) 321-22-55
10	9	2020-11-15 12:00:00.000	100,00	Николаев Артем Викторович	9	Вольская Ирина Чернова	1999-03-25	Пушкина 25	+7 (920) 356-44-25
11	7	2020-11-16 20:00:00.000	200,00	Попов Юрий Александрович	7	Егорова Елена Васильевна	2008-12-22	Горького 16	+7 (910) 456-69-62
12	1	2020-11-26 15:00:00.000	300,00	Попов Юрий Александрович	1	Иванов Иван Иванович	1975-10-25	Пушкина 8	+7 (915) 558-65-69
13	10	2020-11-29 13:00:00.000	200,00	Попов Юрий Александрович	10	Одинцова Ольга Владимировна	2009-03-08	Горького 63	+7 (920) 321-22-55
14	2	2020-11-20 21:00:00.000	300,00	Попов Юрий Александрович	2	Михайлов Дмитрий Владимирович	1980-08-14	Грибоедова 19	+7 (930) 667-56-58
15	4	2020-11-23 10:00:00.000	100,00	Николаев Артем Викторович	4	Чуваев Артем Александрович	2000-09-23	Тимуровцев 15	+7 (910) 999-53-53
16	3	2020-11-16 13:00:00.000	200,00	Филлипов Евгений Геннадиевич	3	Леонов Михаил Максимович	2006-02-05	Васильевская 8	+7 (800) 535-35-35
17	5	2020-11-26 18:00:00.000	300,00	Попов Юрий Александрович	5	Дмитриев Дмитрий Иванович	1975-06-22	Новоселов 35	+7 (992) 321-22-33
18	6	2020-11-30 19:00:00.000	200,00	Филлипов Евгений Геннадиевич	6	Смольский Иван Олегович	2004-10-10	Московский 2	+7 (920) 258-45-65
19	8	2020-11-12 19:00:00.000	300,00	Филлипов Евгений Геннадиевич	8	Олегова Алина Евгеньевна	1990-03-14	Гоголя 23	+7 (932) 999-44-55
20	1	2020-11-11 18:00:00.000	300,00	Попов Юрий Александрович	1	Иванов Иван Иванович	1975-10-25	Пушкина 8	+7 (915) 558-65-69
21	4	2020-11-16 11:00:00.000	100,00	Николаев Артем Викторович	4	Чуваев Артем Александрович	2000-09-23	Тимуровцев 15	+7 (910) 999-53-53
22	8	2020-11-16 12:00:00.000	300,00	Филлипов Евгений Геннадиевич	8	Олегова Алина Евгеньевна	1990-03-14	Гоголя 23	+7 (932) 999-44-55

Рисунок 4.2.3 — Запрос с использованием декартового произведения таблиц

### 3. Многотабличные запросы:

- запрос с использованием соединения двух таблиц по равенству и условием отбора;

```
--вывести ФИО клиентов, ФИО тренеров и дату и время их тренировки
select ФИО, [ФИО инструктора], [Дата и время]
from Клиенты, Тренировки
where Клиенты.[Код клиента]=Тренировки.[Код клиента]
```

	ФИО	ФИО инструктора	Дата и время
1	Иванов Иван Иванович	Попов Юрий Александрович	2020-11-06 12:00:00.000
2	Михайлов Дмитрий Владимирович	Попов Юрий Александрович	2020-11-02 13:00:00.000
3	Чуваев Артем Александрович	Николаев Артем Викторович	2020-11-13 16:00:00.000
4	Леонов Михаил Максимович	Филлипов Евгений Геннадиевич	2020-11-09 18:00:00.000
5	Смольский Иван Олегович	Филлипов Евгений Геннадиевич	2020-11-08 12:00:00.000
6	Егорова Елена Васильевна	Попов Юрий Александрович	2020-11-13 11:00:00.000
7	Дмитриев Дмитрий Иванович	Попов Юрий Александрович	2020-11-26 09:00:00.000
8	Олегова Алина Евгеньевна	Филлипов Евгений Геннадиевич	2020-11-20 20:00:00.000
9	Одинцова Ольга Владимировна	Попов Юрий Александрович	2020-11-19 14:00:00.000
10	Вольская Ирина Чернова	Николаев Артем Викторович	2020-11-15 12:00:00.000
11	Егорова Елена Васильевна	Попов Юрий Александрович	2020-11-16 20:00:00.000
12	Иванов Иван Иванович	Попов Юрий Александрович	2020-11-26 15:00:00.000
13	Одинцова Ольга Владимировна	Попов Юрий Александрович	2020-11-29 13:00:00.000
14	Михайлов Дмитрий Владимирович	Попов Юрий Александрович	2020-11-20 21:00:00.000
15	Чуваев Артем Александрович	Николаев Артем Викторович	2020-11-23 10:00:00.000
16	Леонов Михаил Максимович	Филлипов Евгений Геннадиевич	2020-11-16 13:00:00.000
17	Дмитриев Дмитрий Иванович	Попов Юрий Александрович	2020-11-26 18:00:00.000
18	Смольский Иван Олегович	Филлипов Евгений Геннадиевич	2020-11-30 19:00:00.000
19	Олегова Алина Евгеньевна	Филлипов Евгений Геннадиевич	2020-11-12 19:00:00.000
20	Иванов Иван Иванович	Попов Юрий Александрович	2020-11-11 18:00:00.000
21	Чуваев Артем Александрович	Николаев Артем Викторович	2020-11-16 11:00:00.000
22	Олегова Алина Евгеньевна	Филлипов Евгений Геннадиевич	2020-11-16 12:00:00.000

Рисунок 4.3.1 — Запрос с использованием соединения двух таблиц по равенству и условием отбора

- запрос с использованием соединения трех таблиц по равенству и условием отбора;

```
--вывести ФИО клиентов и названия их абонементов
select distinct ФИО, Название
from Клиенты, Договор, [Тип Абонемента]
where Клиенты.[Код клиента]=Договор.[Код клиента] and Договор.[Код абонемента]=[Тип Абонемента].[Код абонемента]
```

	ФИО	Название
1	Вольская Ирина Чернова	Разовый студенческий
2	Дмитриев Дмитрий Иванович	Разовый взрослый
3	Егорова Елена Васильевна	Разовый детский
4	Иванов Иван Иванович	Разовый взрослый
5	Леонов Михаил Максимович	Разовый детский
6	Михайлов Дмитрий Владимирович	Пятиразовый
7	Михайлов Дмитрий Владимирович	Разовый взрослый
8	Михайлов Дмитрий Владимирович	Разовый детский
9	Одинцова Ольга Владимировна	Разовый детский
10	Олегова Алина Евгеньевна	Разовый взрослый
11	Смольский Иван Олегович	Разовый детский
12	Чуваев Артем Александрович	Разовый студенческий

Рисунок 4.3.2 — Запрос с использованием соединения трех таблиц по равенству и условием отбора

- запрос с использованием соединения по четырем таблицам;

```
--вывести ФИО инструктора, ФИО клиента, номер договора, название абонемента
select distinct [ФИО инструктора], ФИО, [Номер договора], Название
from Тренировки, Договор, Клиенты, [Тип Абонемента]
where Тренировки.[Код клиента]=Договор.[Код клиента] and Договор.[Код клиента]=Клиенты.[Код клиента] and Договор.[Код абонемента]=[Тип Абонемента].[Код абонемента]
```

	ФИО инструктора	ФИО	Номер договора	Название
1	Николаев Артем Викторович	Вольская Ирина Чернова	8	Разовый студенческий
2	Николаев Артем Викторович	Вольская Ирина Чернова	12	Разовый студенческий
3	Николаев Артем Викторович	Чуваев Артем Александрович	3	Разовый студенческий
4	Попов Юрий Александрович	Дмитриев Дмитрий Иванович	4	Разовый взрослый
5	Попов Юрий Александрович	Егорова Елена Васильевна	9	Разовый детский
6	Попов Юрий Александрович	Егорова Елена Васильевна	15	Разовый детский
7	Попов Юрий Александрович	Иванов Иван Иванович	1	Разовый взрослый
8	Попов Юрий Александрович	Иванов Иван Иванович	17	Разовый взрослый
9	Попов Юрий Александрович	Иванов Иван Иванович	20	Разовый взрослый
10	Попов Юрий Александрович	Иванов Иван Иванович	21	Разовый взрослый
11	Попов Юрий Александрович	Иванов Иван Иванович	22	Разовый взрослый
12	Попов Юрий Александрович	Михайлов Дмитрий Владимирович	2	Пятиразовый
13	Попов Юрий Александрович	Михайлов Дмитрий Владимирович	16	Разовый детский
14	Попов Юрий Александрович	Михайлов Дмитрий Владимирович	18	Разовый взрослый
15	Попов Юрий Александрович	Одинцова Ольга Владимировна	7	Разовый детский
16	Попов Юрий Александрович	Одинцова Ольга Владимировна	11	Разовый детский
17	Филлипов Евгений Геннадиевич	Леонов Михаил Максимович	6	Разовый детский
18	Филлипов Евгений Геннадиевич	Леонов Михаил Максимович	19	Разовый детский
19	Филлипов Евгений Геннадиевич	Олегова Алина Евгеньевна	10	Разовый взрослый
20	Филлипов Евгений Геннадиевич	Олегова Алина Евгеньевна	13	Разовый взрослый
21	Филлипов Евгений Геннадиевич	Смольский Иван Олегович	5	Разовый детский
22	Филлипов Евгений Геннадиевич	Смольский Иван Олегович	14	Разовый детский

Рисунок 4.3.3 — Запрос с использованием соединения по четырем таблицам

- запрос с использованием левого внешнего соединения;

```
--вывести ФИО всех клиентов и их тренеров, если имеется
```

```

select distinct ФИО as Клиент, [ФИО инструктора] as Тренер
from Клиенты left join Тренировки on Клиенты.[Код клиента]=Тренировки.[Код клиента]

```

	Клиент	Тренер
1	Вольская Ирина Чернова	Николаев Артем Викторович
2	Дмитриев Дмитрий Иванович	Попов Юрий Александрович
3	Егорова Елена Васильевна	Попов Юрий Александрович
4	Житков Ильдар Викторович	NULL
5	Иванов Иван Иванович	Попов Юрий Александрович
6	Леонов Михаил Максимович	Филлипов Евгений Геннадиевич
7	Михайлов Дмитрий Владимирович	Попов Юрий Александрович
8	Одинцова Ольга Владимировна	Попов Юрий Александрович
9	Олегова Алина Евгеньевна	Филлипов Евгений Геннадиевич
10	Смольский Иван Олегович	Филлипов Евгений Геннадиевич
11	Чуваев Артем Александрович	Николаев Артем Викторович

Рисунок 4.3.4 — Запрос с использованием левого внешнего соединения

- запрос на использование правого внешнего соединения;

```

--вывести ФИО всех клиентов и их тренеров, если имеется
select distinct ФИО as Клиент, [ФИО инструктора] as Тренер
from Тренировки right join Клиенты on Клиенты.[Код клиента]=Тренировки.[Код клиента]

```

	Клиент	Тренер
1	Вольская Ирина Чернова	Николаев Артем Викторович
2	Дмитриев Дмитрий Иванович	Попов Юрий Александрович
3	Егорова Елена Васильевна	Попов Юрий Александрович
4	Житков Ильдар Викторович	NULL
5	Иванов Иван Иванович	Попов Юрий Александрович
6	Леонов Михаил Максимович	Филлипов Евгений Геннадиевич
7	Михайлов Дмитрий Владимирович	Попов Юрий Александрович
8	Одинцова Ольга Владимировна	Попов Юрий Александрович
9	Олегова Алина Евгеньевна	Филлипов Евгений Геннадиевич
10	Смольский Иван Олегович	Филлипов Евгений Геннадиевич
11	Чуваев Артем Александрович	Николаев Артем Викторович

Рисунок 4.3.5 – Запрос на использование правого внешнего соединения

- запрос с использованием функции COUNT;

```

--вывести ФИО клиента и количество его тренировок
select ФИО, COUNT (Тренировки.[Код клиента]) as [Кол-во тренировок]
from Клиенты, Тренировки
where Клиенты.[Код клиента]=Тренировки.[Код клиента]
group by Клиенты.ФИО

```

	ФИО	Кол-во тренировок
1	Вольская Ирина Чернова	1
2	Дмитриев Дмитрий Иванович	2
3	Егорова Елена Васильевна	2
4	Иванов Иван Иванович	3
5	Леонов Михаил Максимович	2
6	Михайлов Дмитрий Владимирович	2
7	Одинцова Ольга Владимировна	2
8	Олегова Алина Евгеньевна	3
9	Смольский Иван Олегович	2
10	Чуваев Артем Александрович	3

Рисунок 4.3.6 – запрос с использованием функции COUNT

- запрос с использованием функции SUM;

```
--Вывести фио клиента и сколько он потратил на тренировки
select ФИО, Sum (Тренировки.[Стоимость]) as [Стоимость тренировок]
from Клиенты, Тренировки
where Клиенты.[Код клиента]=Тренировки.[Код клиента]
group by Клиенты.ФИО
```

	ФИО	Стоимость тренировок
1	Вольская Ирина Чернова	100,00
2	Дмитриев Дмитрий Иванович	600,00
3	Егорова Елена Васильевна	400,00
4	Иванов Иван Иванович	900,00
5	Леонов Михаил Максимович	400,00
6	Михайлов Дмитрий Владимирович	600,00
7	Одинцова Ольга Владимировна	400,00
8	Олегова Алина Евгеньевна	900,00
9	Смольский Иван Олегович	500,00
10	Чуваев Артем Александрович	300,00

Рисунок 4.3.7 – Запрос с использованием функции SUM

- 3 запроса с использованием временных функций;

```
--вывести несовершеннолетних клиентов
select *
from Клиенты
where Year([Дата рождения]) > 2002
```



	Код клиента	ФИО	Дата рождения	Адрес	Телефон
1	3	Леонов Михаил Максимович	2006-02-05	Васильевская 8	+7 (800) 535-35-35
2	6	Смольский Иван Олегович	2004-10-10	Московский 2	+7 (920) 258-45-65
3	7	Егорова Елена Васильевна	2008-12-22	Горького 16	+7 (910) 456-69-62
4	10	Одинцова Ольга Владимировна	2009-03-08	Горького 63	+7 (920) 321-22-55

Рисунок 4.3.8 – Запрос с функцией year()

```
--вывести номера договоров, заключенных в январе
select [Номер договора]
from Договор
where Month( [Дата и время покупки])=1
```

	Номер договора
1	1

Рисунок 4.3.9 – Запрос с функцией month()

```
select DAY([Дата и время покупки]) as [день заключения договора]
from Договор
where [Код клиента]=(select Клиенты.[Код клиента] from Клиенты where ФИО ='Леонов
Михаил Максимович')
```

	день заключения договора
1	18
2	10

Рисунок 4.3.10 – Запрос с функцией day()

- запрос с использованием группировки по одному столбцу;

```
--вывести тренеров, кол-во клиентов
select [ФИО инструктора], COUNT ([Код клиента]) as [Кол-во клиентов]
from Тренировки
group by [ФИО инструктора]
```

	ФИО инструктора	Кол-во клиентов
1	Николаев Артем Викторович	4
2	Попов Юрий Александрович	11
3	Филлипов Евгений Геннадиевич	7

Рисунок 4.3.11 — Запрос с использованием группировки по одному столбцу

- запрос на использование группировки по нескольким столбцам;

```
select [ФИО инструктора], ФИО, COUNT(*) as [Кол-во занятий]
from Тренировки inner join Клиенты on Тренировки.[Код клиента]=Клиенты.[Код клиента]
group by [ФИО инструктора], ФИО
```



	ФИО инструктора	ФИО	Кол-во занятий
1	Николаев Артем Викторович	Вольская Ирина Чернова	1
2	Попов Юрий Александрович	Дмитриев Дмитрий Иванович	2
3	Попов Юрий Александрович	Егорова Елена Васильевна	2
4	Попов Юрий Александрович	Иванов Иван Иванович	3
5	Филлипов Евгений Геннадиевич	Леонов Михаил Максимович	2
6	Попов Юрий Александрович	Михайлов Дмитрий Владимирович	2
7	Попов Юрий Александрович	Одинцова Ольга Владимировна	2
8	Филлипов Евгений Геннадиевич	Олегова Алина Евгеньевна	3
9	Филлипов Евгений Геннадиевич	Смольский Иван Олегович	2
10	Николаев Артем Викторович	Чуваев Артем Александрович	3

Рисунок 4.3.12 — Запрос на использование группировки по нескольким столбцам

- запрос с использованием условия отбора групп HAVING;

```
-- вывести клиентов, которые потратили на тренировки не менее 500
select Клиенты.ФИО, SUM (Тренировки.Стоимость) As Потратили
from Клиенты, Тренировки
where Клиенты.[Код клиента]=Тренировки.[Код клиента]
group by Клиенты.ФИО
having SUM(Тренировки.Стоимость)>=500
```

	ФИО	Потратили
1	Дмитриев Дмитрий Иванович	600,00
2	Иванов Иван Иванович	900,00
3	Михайлов Дмитрий Владимирович	600,00
4	Олегова Алина Евгеньевна	900,00
5	Смольский Иван Олегович	500,00

Рисунок 4.3.13 — Запрос с использованием условия отбора групп HAVING

- запрос с использованием сортировки по столбцу;

```
--Вывести фио клиентов в порядке убывания потраченного на тренировки
select Клиенты.ФИО, SUM (Тренировки.Стоимость) As Потратили
from Клиенты, Тренировки
where Клиенты.[Код клиента]=Тренировки.[Код клиента]
group by Клиенты.ФИО
order by Потратили DESC
```

	ФИО	Потратили
1	Иванов Иван Иванович	900,00
2	Олегова Алина Евгеньевна	900,00
3	Михайлов Дмитрий Владимирович	600,00
4	Дмитриев Дмитрий Иванович	600,00
5	Смольский Иван Олегович	500,00
6	Одинцова Ольга Владимировна	400,00
7	Егорова Елена Васильевна	400,00
8	Леонов Михаил Максимович	400,00
9	Чуваев Артем Александрович	300,00
10	Вольская Ирина Чернова	100,00

Рисунок 4.3.14 — Запрос с использованием сортировки по столбцу

#### 4. Запросы на модификацию таблиц:

- запрос на добавление новых данных в таблицу;

```
--добавить новый абонемент
insert into [Тип Абонемента]
values
('Разовый школьный', 'Одно посещение школьника', 150, '2020-12-31')
```

```
(затронута одна строка)

Время выполнения: 2020-12-10T11:25:11.8732948+03:00
```

Рисунок 4.4.1 — Запрос на добавление новых данных в таблицу

- запрос на добавление новых данных по результатам выполненного запроса;

```
insert into Договор
select [Код клиента], [Код абонемента]+1, [Дата и время покупки]
from Договор
where [Номер договора]=22
```

	Номер догов...	Код клиента	Код абонемен...	Дата и время ...
►	1	1	2	2020-01-20 12:3...
	2	2	4	2020-10-30 13:1...
	3	4	3	2020-10-22 10:3...
	4	5	2	2020-10-23 09:4...
	5	6	1	2020-10-11 14:5...
	6	3	1	2020-10-18 19:2...
	7	10	1	2020-10-23 20:2...
	8	9	3	2020-10-24 20:0...
	9	7	1	2020-10-13 19:0...
	10	8	2	2020-10-09 18:1...
	11	10	1	2020-10-27 21:1...
	12	9	3	2020-10-11 20:5...
	13	8	2	2020-10-30 11:1...
	14	6	1	2020-10-21 10:2...
	15	7	1	2020-10-14 21:1...
	16	2	1	2020-10-04 22:3...
	17	1	2	2020-10-10 13:3...
	18	2	2	2020-10-10 13:3...
	19	3	1	2020-10-10 13:3...
	20	1	2	2020-10-20 12:3...
	21	1	2	2020-10-20 12:3...
	22	1	2	2020-10-20 12:3...
	23	1	3	2020-10-20 12:3...
*	NULL	NULL	NULL	NULL

Рисунок 4.4.2 — Запрос на добавление новых данных по результатам выполненного запроса

- запрос на обновление существующих данных в таблице;

--увеличить стоимость всех абонементов

update [Тип Абонемента]

set Стоимость = Стоимость + 100

	Код абонемен...	Название	Описание	Стоимость	Срок действия
►	1	Разовый детск...	Одно посещен...	300,0000	2020-12-31
	2	Разовый взрос...	Одно посещен...	400,0000	2020-12-31
	3	Разовый студе...	Одно посещен...	200,0000	2020-12-31
	4	Пятиразовый	Пять посещен...	900,0000	2020-12-31
	5	Месячный взр...	Месяц без огр...	1600,0000	2021-12-31
	6	Месячный сту...	Месяц без огр...	1100,0000	2021-12-31
	7	Пятиразовый с...	Пять посещен...	600,0000	2021-12-31
	8	Трехмесячный	Пять месяцев ...	3100,0000	2021-12-31
	9	Полугодовой	Полгода без ог...	5100,0000	2022-12-31
	10	Трехмесячный...	Три месяца сту...	2000,0000	2021-12-31
	11	Разовый школ...	Одно посещен...	250,0000	2020-12-31
*	NULL	NULL	NULL	NULL	NULL

Рисунок 4.4.3 — Запрос на обновление существующих данных в таблице

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

```
--скидка 30% на взрослый абонемент
update [Тип Абонемент]
set Стоимость = Стоимость * 0.7
where [Код абонемент] = (select [Код абонемент]
                        From [Тип Абонемент]
                        where Название='Разовый взрослый')
```

	Код абонемен...	Название	Описание	Стоимость	Срок действия
►	1	Разовый детск...	Одно посещен...	300,0000	2020-12-31
	2	Разовый взрос...	Одно посещен...	196,0000	2020-12-31
	3	Разовый студе...	Одно посещен...	200,0000	2020-12-31
	4	Пятиразовый	Пять посещен...	900,0000	2020-12-31
	5	Месячный взр...	Месяц без огр...	1600,0000	2021-12-31
	6	Месячный сту...	Месяц без огр...	1100,0000	2021-12-31
	7	Пятиразовый с...	Пять посещен...	600,0000	2021-12-31
	8	Трехмесячный	Пять месяцев ...	3100,0000	2021-12-31
	9	Полугодовой	Полгода без ог...	5100,0000	2022-12-31
	10	Трехмесячный...	Три месяца сту...	2000,0000	2021-12-31
	11	Разовый школ...	Одно посещен...	250,0000	2020-12-31
*	NULL	NULL	NULL	NULL	NULL

Рисунок 4.4.4 — Запрос на обновление существующих данных по результатам подзапроса во фразе WHERE