**Федеральное государственное бюджетное образовательное учреждение высшего образования «РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА И ГОСУДАРСТВЕННОЙ СЛУЖБЫ ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет: Управление

Кафедра: Информатики и информационных технологий

Направление подготовки: 09.03.03 Прикладная информатика

Направленность: Корпоративные информационные системы управления

**Отчет о проделанной работе**

По дисциплине: Базы данных

На тему: «Разработка базы данных для фитнес-клуба»

**АВТОР**

Обучающийся 2 курса группы ИБ-321

Очной формы обучения

\_\_\_\_\_\_\_\_\_\_\_\_\_ Наумов А. Е.

**НАУЧНЫЙ РУКОВОДИТЕЛЬ**

Старший преподаватель

Оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 20\_\_\_\_ г.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Миронов Д.Ю.

Нижний Новгород, 2022г.

**Оглавление**

[**Введение** 3](#_Toc121683736)

[**Глава 1. Основные понятия** 4](#_Toc121683737)

[**Глава 2. Описание предметной области, процессов и объектов БД** 7](#_Toc121683738)

[**Глава 3. Схемы данных на логическом и физическом уровнях проектирования и нормализация БД** 10](#_Toc121683739)

[**Глава 4. Запросы.** 13](#_Toc121683740)

[4.1. Виды запросов. 13](#_Toc121683741)

[4.2. Виды объединений. 13](#_Toc121683742)

[4.3 Текст запросов. 13](#_Toc121683743)

[**Заключение** 23](#_Toc121683744)

# **Введение**

В настоящее время в повседневной практике для принятия решений приходится использовать большие объемы разнообразной информации, которую необходимо собрать, соответствующим образом расклассифицировать, организовать её хранение и, наконец, использовать. Время использования картотек на бумажных носителях или журналов регистрации давно прошло, и сейчас для этих целей широко используются различные базы данных (БД), находящихся на компьютерах. Использование баз данных становится неотъемлемой составляющей деловой деятельности современного человека и функционирования преуспевающих организаций.

**Целью** контрольной работы является разработка базы данных для автоматизации работы **фитнес-клуба.**

Для достижения данной цели необходимо выполнить следующие **задачи**:

1. Выявить основные понятия.

2. Описать выбранную предметную область, процессы и объекты, которые будут хранить в нашей БД.

3. Изобразить схему данных на логическом и физическом уровнях проектирования и доказать, что все таблицы удовлетворяют 3-ей нормальной форме.

4. Показать работу всех видов запросов, объединений, подзапросов, производных таблиц, автовычисляемых полей.

Практическая значимость данной работы заключается в том, что в результате проведенных работ, будет создана готовая БД, которую уже можно использовать в Фитнес-клубе.

# **Глава 1. Основные понятия**

**База данных (БД)** - совокупность данных, структурированных таким образом, чтобы было возможно осуществлять хранение, поиск и обработку этих данных.

**Система управления базами данных (СУБД)** – совокупность специальных программ, которая обеспечивает управление созданием и использованием баз данных.

**Модель** – приблизительная копия реального объекта или явления, отражающая его существенные свойства для данного исследования.

**Модель данных** – совокупность определения данных и операций по их обработке, в которой объекты позволяют описывать структуру данных, а операторы обеспечивают изменение данных.

**Реляционная модель данных (РМД)** – прикладная теория проектирования баз данных, основой в которой являются отношения между объектами, а сами объекты являются абстрактными моделями.

Реляционная модель данных предъявляет следующие требования к каждой таблице РБД:

1. Отсутствие кортежей-дубликатов.

2. Отсутствие упорядоченности кортежей.

3. Отсутствие упорядоченности атрибутов.

4. Атомарность (неделимость) значений атрибутов

**Ключ** – столбец или минимальный набор столбцов, однозначно определяющий каждую строку таблицы. Если ключ состоит из одного столбца, то его называют простым, а если из нескольких, то составным. Ключ, который определяет уникальность каждой записи внутри таблицы, называют первичным ключом (**primary key, PK**). Связи между таблицами как раз и устанавливаются на основании первичных ключей, т.е. в связываемой таблице создается копия первичного ключа с тем же типом данных, при этом имя этого столбца или столбцов (если ключ составной) может быть любым, данная копия первичного ключа называется внешним ключом (**foreign key, FK**).

В реляционных база данных существуют *три вида отношений* между таблицами:

**1. Отношение один к одному.**

Данный вид связи предполагает, что каждая запись в одной таблице связана не более чем с одной записью в другой таблице. Данный вид связи обычно применяется к тем свойствам объектов, которые могут отсутствовать и присутствовать с равной вероятностью, например, наличие автомобиля у человека.

**2. Отношение один ко многим.**

Это один из наиболее распространенных видов связи при проектировании БД, поскольку все объекты в реальном мире имеют множественные связи. Данный вид связи предполагает, что каждая запись в одной таблице связана с произвольным количеством записей в другой. Записи же второй таблицы связаны не более чем с одной записью первой таблицы.

**3. Отношение многие ко многим.**

Это также очень распространенный вид связи по тем же причинам. Данный вид связи предполагает, что произвольное количество записей в одной таблице связано с произвольным количеством записей в другой. Такой вид связи означает наличие третьей таблицы – таблицы посредника, связанного один ко многим с каждой из двух таблиц.

**Нормальная форма (NF)** – требование, которому должна удовлетворять таблица. Процесс преобразования схемы базы данных к виду, отвечающему требованиям нормальных форм, называется нормализацией, а сознательное или несознательное нарушение этих требований денормализацией.

Под **нормализацией БД** понимают процесс преобразования БД к виду, которые будет удовлетворять нормальным формам. Нам необходимо проверить, все ли таблицы в БД находятся в третьей нормальной форме.

**Первая нормальная форма(1NF)** – на пересечении строки и столбца должны находиться атомарные элементы;

В каждой из таблиц на пересечении строки и столбца находятся атомарные элементы. Значит, выполняется 1NF.

**Вторая нормальная форма(2NF)** – необходимо выполнение 1NF и каждый не ключевой столбец зависит от всего составного ключа и не зависит ни от одной его части. Если у таблицы отсутствует составной ключ, она также находится в 2NF;

В таблицах нет составных ключей, значит, все они удовлетворяют 2NF**.**

**Третья нормальная форма(3NF)** – выполнение 1NF и 2NF и любой не ключевой атрибут не зависит от другого не ключевого атрибута;

Ни одни не ключевые атрибуты не зависят от других не ключевых атрибутов. Значит, все таблицы удовлетворяют 3NF.

# **Глава 2. Описание предметной области, процессов и объектов БД**

Главной задачей контрольной работы является разработка базы данных **фитнес-клуба** с помощью СУБД MS SQL Server. Данная база будет содержать данные об абонементах, работниках и клиентах. Клиент может владеть несколькими абонементами, быть включен в группы, брать индивидуальные занятия. Для групп составлено расписание, в котором учитываются помещения и их характеристики. Клиентами так же могут быть фирмы и оплачивать абонементы своих сотрудников. Тренера могут вести несколько групп. В данной БД хранятся персональные данные сотрудников и клиентов.

***Исходя из задач туристического агентства, необходимо создать следующие таблицы:***

* 1. Группа
  2. Зал
  3. Тренер
  4. Индивидуальные тренировки
  5. Абонементы групп
  6. Расписание
  7. Сотрудник
  8. Клиент
  9. Вид абонемента
  10. Абонемент
  11. Корпоративный клиент

Для того чтобы описать процессы и объекты, которые будут храниться в БД, рассмотрим подробнее каждую таблицу.

1. Таблица **«Группа»** включает в себя такие столбцы как:

* Название
* Дисциплина

В этой таблице будет храниться информация о группе.

2. Таблица **«Зал»** включает в себя такие столбцы как:

* Название
* Площадь
* Напольное покрытие

В этой таблице будет храниться информация о залах.

3. Таблица **«Тренер»** включает в себя такие столбцы как:

* Количество индивидуальных тренировок

В этой таблице будет храниться информация о тренере.

4. Таблица **«Индивидуальная тренировка»** включает в себя такие столбцы как:

* Дисциплина
* Цена
* Дата
* Время начала
* Время конца

В этой таблице будет храниться информация об индивидуальной тренировке.

5. Таблица **«Абонементы групп»** включает в себя только служебные данные не несущие информационной нагрузки.

6. Таблица **«Расписание»** включает в себя такие столбцы как:

* Дата
* Время начала
* Время конца

В этой таблице будет храниться информация о расписании групповых занятий.

7. Таблица **«Сотрудник»** включает в себя такие столбцы как:

* Имя
* Фамилия
* Отчество
* Серия паспорта
* Номер паспорта
* Телефон
* Адрес
* СНИЛС
* Должность
* Квалификация
* Оклад  
  В этой таблице будет храниться информация о сотрудниках.

8. Таблица **«Клиент»** включает в себя такие столбцы как:

* Имя
* Фамилия
* Отчество
* Телефон
* Серия паспорта
* Номер паспорта

В этой таблице будет храниться информация о клиентах.

9. Таблица **«Вид абонемента»** включает в себя такие столбцы как:

* Название
* Цена
* Конец срока – это какое время действует абонемент, 1 месяц, 1 год и т.д.
* Цена
* Время начала – это время, со скольки можно посещать фитнес-клуб
* Время конца – это время, до скольки можно посещать фитнес-клуб

В этой таблице будет храниться информация об условиях каждого вида абонемента.

10. Таблица **«Абонемента»** включает в себя такие столбцы как:

* Дата получения

В этой таблице будет храниться информация об дате получения абонемента.

11. Таблица **«Корпоративный клиент»** включает в себя такие столбцы как:

* Название фирмы
* Адрес
* Контактный номер телефона
* Банковский счет

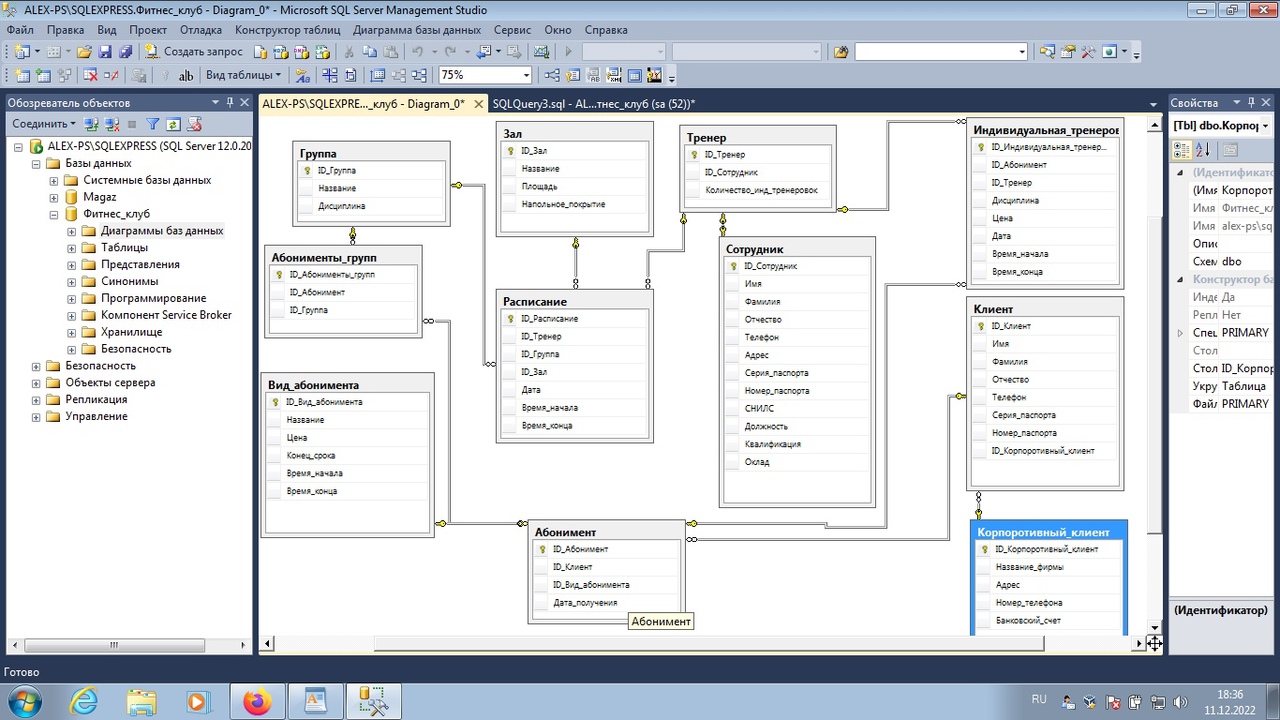
В этой таблице будет храниться информация о корпоративном клиенте.

# **Глава 3. Схемы данных на логическом и физическом уровнях проектирования и нормализация БД**

В БД будут существовать такие таблицы как:

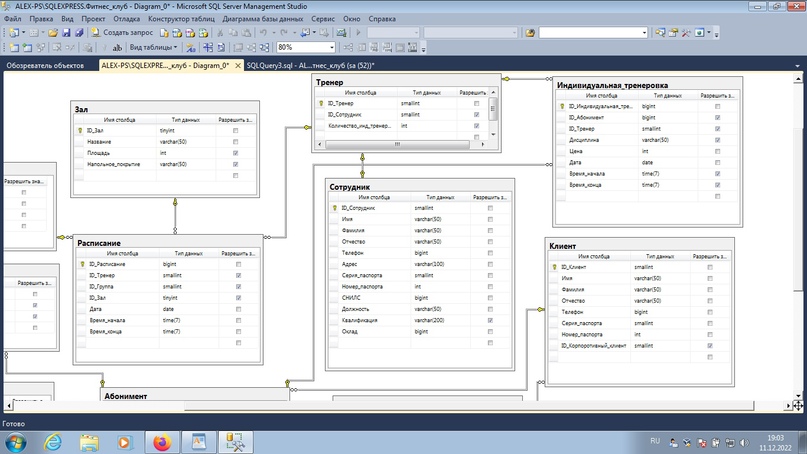
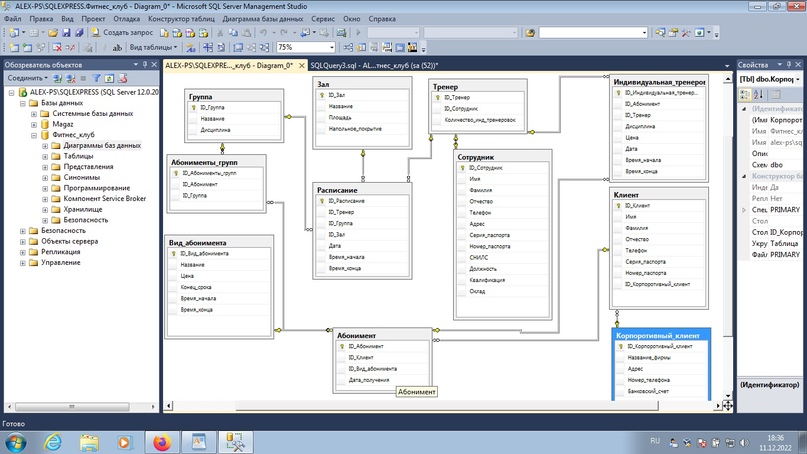
* 1. Группа
  2. Зал
  3. Тренер
  4. Индивидуальные тренировки
  5. Абонементы групп
  6. Расписание
  7. Сотрудник
  8. Клиент
  9. Вид абонемента
  10. Абонемент
  11. Корпоративный клиент

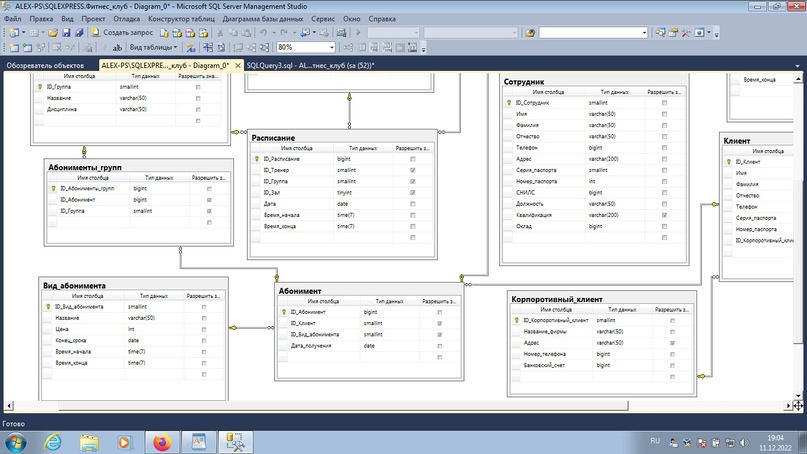
Схемы данных мы будем строить с помощью MS SQL Server диаграмм БД.



*Схема данных на логическом уровне проектирования.*

Далее необходимо построить схему данных на **физическом уровне** проектирования.





*Схема данных на физическом уровне проектирования.*

Таким образом, мы построили схемы данных на логическом и физическом уровне проектирования БД с помощью MS SQL Server диаграмм БД. При проектировании важно помнить, что внешние ключи и соответствующие им первичные ключи должны иметь одинаковый тип данных.

Проверим, соответствие нашей базы на удовлетворение 3NF. В каждой таблице на пересечении строки и столбца находятся атомарные элементы. Следовательно, соответствие 1NF выполнено. В таблицах отсутствуют составные ключи, поэтому все не ключевые атрибуты зависят от первичного ключа в целом. 2NF выполняется. Не ключевые столбцы не зависят друг от друга. 3NF также удовлетворена.

Наша база находится в 3-ей нормальной форме, чего достаточно для устойчивой работы.

# **Глава 4. Запросы.**

## 4.1. Виды запросов.

Запросы – это специальные средства выбора нужной информации из БД.

Запросы бывают 4х видов:

* На выборку – Select;
* На добавление – Insert;
* На обновление – Update;
* На удаление – Delete.

## 4.2. Виды объединений.

Далее рассмотрим, как можно соединять таблицы. С помощью JOIN оператора можно соединять несколько таблиц. Соединения могут быть внутренними(INNER), внешними(OUTER), правыми(RIGHT) или левыми(LEFT) внешними и полными(FULL).

1. Соединение FULL JOIN
2. Соединение OUTER JOIN
3. Соединение INNER JOIN

## 4.3 Текст запросов.

CREATE DATABASE Фитнес\_клуб

GO

USE Фитнес\_клуб

GO

CREATE TABLE Корпоротивный\_клиент (

ID\_Корпоротивный\_клиент smallint identity(1,1) NOT NULL PRIMARY KEY,

Название\_фирмы varchar(50) not null,

Адрес varchar(50) null,

Номер\_телефона bigint not null,

Банковский\_счет bigint not null

)

CREATE TABLE Группа (

ID\_Группа smallint identity(1,1) NOT NULL PRIMARY KEY,

Название varchar(50) not null,

Дисциплина varchar(50) not null

)

CREATE TABLE Вид\_абонимента (

ID\_Вид\_абонимента smallint identity(1,1) NOT NULL PRIMARY KEY,

Название varchar(50) not null,

Цена int not null,

Конец\_срока date not null,

Время\_начала time not null,

Время\_конца time not null

)

CREATE TABLE Зал (

ID\_Зал tinyint identity(1,1) NOT NULL PRIMARY KEY,

Название varchar(50) not null,

Площадь int null,

Напольное\_покрытие varchar(50) null

)

CREATE TABLE Сотрудник (

ID\_Сотрудник smallint identity(1,1) NOT NULL PRIMARY KEY,

Имя varchar(50) not null,

Фамилия varchar(50) not null,

Отчество varchar(50) not null,

Телефон bigint not null,

Адрес varchar(100) not null,

Серия\_паспорта smallint not null,

Номер\_паспорта int not null,

СНИЛС bigint not null,

Должность varchar(50) not null,

Квалификация varchar(200) null,

Оклад bigint not null

)

go

CREATE TABLE Клиент (

ID\_Клиент smallint identity(1,1) NOT NULL PRIMARY KEY,

Имя varchar(50) not null,

Фамилия varchar(50) not null,

Отчество varchar(50) not null,

Телефон bigint not null,

Серия\_паспорта smallint not null,

Номер\_паспорта int not null,

ID\_Корпоротивный\_клиент smallint null FOREIGN KEY REFERENCES Корпоротивный\_клиент(ID\_Корпоротивный\_клиент)

)

CREATE TABLE Тренер (

ID\_Тренер smallint identity(1,1) NOT NULL PRIMARY KEY,

ID\_Сотрудник smallint UNIQUE FOREIGN KEY REFERENCES Сотрудник(ID\_Сотрудник),

Количество\_инд\_тренеровок int null

)

go

CREATE TABLE Расписание(

ID\_Расписание bigint identity(1,1) NOT NULL PRIMARY KEY,

ID\_Тренер smallint FOREIGN KEY REFERENCES Тренер(ID\_Тренер),

ID\_Группа smallint FOREIGN KEY REFERENCES Группа(ID\_Группа),

ID\_Зал tinyint FOREIGN KEY REFERENCES Зал(ID\_Зал),

Дата date Not Null,

Время\_начала time not null,

Время\_конца time not null

)

CREATE TABLE Абонимент(

ID\_Абонимент bigint identity(1,1) NOT NULL PRIMARY KEY,

ID\_Клиент smallint FOREIGN KEY REFERENCES Клиент(ID\_Клиент),

ID\_Вид\_абонимента smallint FOREIGN KEY REFERENCES Вид\_абонимента(ID\_Вид\_абонимента),

Дата\_получения date not null

)

go

CREATE TABLE Абонименты\_групп(

ID\_Абонименты\_групп bigint identity(1,1) NOT NULL PRIMARY KEY,

ID\_Абонимент bigint FOREIGN KEY REFERENCES Абонимент(ID\_Абонимент),

ID\_Группа smallint FOREIGN KEY REFERENCES Группа(ID\_Группа)

)

CREATE TABLE Индивидуальная\_тренеровка (

ID\_Индивидуальная\_тренеровка bigint identity(1,1) NOT NULL PRIMARY KEY

)

go

ALTER TABLE Индивидуальная\_тренеровка ADD ID\_Абонимент bigint FOREIGN KEY REFERENCES Абонимент(ID\_Абонимент),

ID\_Тренер smallint FOREIGN KEY REFERENCES Тренер(ID\_Тренер),

Дисциплина varchar(50),

Цена int not null,

Дата date not null,

Время\_начала time,

Время\_конца time

go

INSERT INTO Вид\_абонимента (Название, Цена, Конец\_срока, Время\_начала, Время\_конца)

VALUES ('Студенческое утро', 1500, '0001-02-01','06:00:00','14:00:00'),

('Студенческое утро', 1500, '0001-02-01','06:00:00','14:00:00'),

('Студенческое утро', 1500, '0001-02-01','06:00:00','14:00:00'),

('Студенческое утро', 1500, '0001-02-01','06:00:00','14:00:00'),

('Студенческое утро', 10000, '0001-09-01','06:00:00','14:00:00'),

('Студенческое утро', 15000, '0002-01-01','06:00:00','14:00:00'),

('Студенческий FULL', 2000, '0001-02-01','06:00:00','23:00:00'),

('Студенческий FULL', 13000, '0001-09-01','06:00:00','23:00:00'),

('Студенческий FULL', 20000, '0002-01-01','06:00:00','23:00:00'),

('Стрейчинг', 2500, '0001-02-01','06:00:00','23:00:00'),

('Детский бокс', 2000, '0001-02-01','06:00:00','23:00:00'),

('Детское Самбо', 2000, '0001-02-01','06:00:00','23:00:00'),

('Свободное плаванье', 2000, '0001-02-01','06:00:00','23:00:00'),

('Свободное плаванье', 16000, '0002-01-01','06:00:00','23:00:00'),

('Разовая', 500, '0001-01-02','06:00:00','23:00:00'),

('Тяжелая атлетика', 3000, '0001-02-01','06:00:00','23:00:00'),

('Классический', 2500, '0001-02-01','06:00:00','23:00:00'),

('Классический', 30000, '0002-01-01','06:00:00','23:00:00'),

('Классический утро', 1500, '0001-02-01','06:00:00','10:00:00'),

('Борьба для взрослых', 3000, '0001-02-01','06:00:00','23:00:00'),

('Бокс для взрослых', 3000, '0001-02-01','06:00:00','23:00:00')

go

DELETE FROM Вид\_абонимента

WHERE ID\_Вид\_абонимента not in (

SELECT MIN(ID\_Вид\_абонимента)

FROM Вид\_абонимента

GROUP BY Название, Цена, Конец\_срока, Время\_начала, Время\_конца

)

INSERT INTO Сотрудник (Имя, Фамилия, Отчество, Телефон, Адрес,Серия\_паспорта, Номер\_паспорта, СНИЛС, Должность, Оклад)

VALUES ('Петр','Петров','Петрович', 89601234567, 'ул. Пушкина 7-53', 2217, 123456, 12345678910, 'Директор по продажам', 55000),

('Иван','Иванов','Иванович', 89601234568, 'ул. Лесная 7-54', 2217, 123457, 12345678911, 'Заведующий хозяйством', 45000),

('Мария','Орлова','Евгеньевна', 89601234569, 'ул. Алексеева 33-3', 2217, 123458, 12345678912, 'Уборщица', 25000),

('Дарья','Цветкова','Константиновна', 89601234563, 'ул. Перьмякова 46-3', 2217, 123453, 12345678913, 'Бухгалтер', 45000),

('Алина','Полякова','Евгеньевна', 89601234561, 'ул. Мухина 33-3', 2217, 123258, 12345628912, 'Тренер', 30000),

('Алексей','Кашин','Максимович', 89601334561, 'ул. Гагарина 34-3', 2217, 123358, 12345608912, 'Тренер', 30000),

('Василий','Васин','Васильевич', 89601234061, 'ул. Махина 73-3', 2217, 123268, 12340628912, 'Тренер', 30000),

('Евгений','Попов','Егорьевич', 89441234561, 'ул. Мухина 33-3', 2217, 103258, 12685628912, 'Тренер', 30000)

INSERT INTO Зал (Название, Площадь, Напольное\_покрытие)

VALUES ('Детский бассейн', 100, null),

('Бассейн', 500, null),

('Борцовский зал', 300, 'Татами'),

('Боксерский зал', 200, 'Паркет'),

('Зал для танцев', 90, 'Паркет'),

('Универсальный зал ', 100, 'Мягкое покрытие'),

('Тренажерный зал №1', 80, 'Резиновое покрытие'),

('Тренажерный зал №2', 200, 'Резиновое покрытие')

INSERT INTO Группа (Название, Дисциплина)

VALUES ('Бокс 7-12 лет', 'Бокс'),

('Бокс 13-16 лет', 'Бокс'),

('Бокс +17 лет', 'Бокс'),

('Самбо 7-15 лет', 'Самбо'),

('Самбо +16 лет', 'Самбо'),

('Румба для взрослых', 'Танцы'),

('Самба для взрослых', 'Танцы'),

('Плаванье 5-12 лет', 'Плаванье'),

('Плаванье +12 лет', 'Плаванье')

INSERT INTO Корпоротивный\_клиент (Название\_фирмы, Номер\_телефона, Банковский\_счет, Адрес)

VALUES ('СтройИнвест', 89601882266, 1234123412341234, null),

('ITlab', 89601882446, 1234123412341235, 'Веденяпина 12-13'),

('ГАЗ', 89699882446, 1234123412341236, 'проспект Ленина 5')

go

INSERT INTO Клиент(Имя, Фамилия, Отчество, телефон, Серия\_паспорта, Номер\_паспорта, ID\_Корпоротивный\_клиент)

VALUES ('Имя1', 'Фамилия1', 'Отчество1', 89601112233, 2217, 123456, 1),

('Имя2', 'Фамилия2', 'Отчество2', 89602112233, 2217, 122456, 1),

('Имя3', 'Фамилия3', 'Отчество3', 89601122233, 2217, 123256, 1),

('Имя4', 'Фамилия4', 'Отчество4', 89604112233, 2217, 143456, 1),

('Имя5', 'Фамилия5', 'Отчество5', 89601112235, 2217, 153456, 2),

('Имя6', 'Фамилия6', 'Отчество6', 89661112233, 2217, 623456, 2),

('Имя7', 'Фамилия7', 'Отчество7', 89601112237, 2217, 723456, 3),

('Имя8', 'Фамилия8', 'Отчество8', 89608112233, 2217, 823456, 3),

('Имя9', 'Фамилия9', 'Отчество9', 89601112239, 2217, 923456, 3),

('Имя10', 'Фамилия10', 'Отчество10', 89600112233, 2217, 103456, 3),

('Имя11', 'Фамилия11', 'Отчество11', 89601112133, 2217, 123416, null),

('Имя12', 'Фамилия12', 'Отчество12', 89602112233, 2217, 122256, null),

('Имя13', 'Фамилия13', 'Отчество13', 89603312233, 2217, 123436, null),

('Имя14', 'Фамилия14', 'Отчество14', 89604412233, 2217, 144456, null),

('Имя15', 'Фамилия15', 'Отчество15', 89601155233, 2217, 123556, null),

('Имя16', 'Фамилия16', 'Отчество16', 89601116633, 2217, 663456, null),

('Имя17', 'Фамилия17', 'Отчество17', 89601177733, 2217, 127756, null),

('Имя18', 'Фамилия18', 'Отчество18', 89688812233, 2217, 883456, null),

('Имя19', 'Фамилия19', 'Отчество19', 89601112299, 2217, 123996, null),

('Имя21', 'Фамилия21', 'Отчество21', 89111112233, 2217, 111456, null),

('Имя22', 'Фамилия22', 'Отчество22', 89622212233, 2217, 122226, null),

('Имя23', 'Фамилия23', 'Отчество23', 89603312233, 2217, 323456, null),

('Имя24', 'Фамилия24', 'Отчество24', 89601144433, 2217, 124454, null),

('Имя25', 'Фамилия25', 'Отчество25', 89551112233, 2217, 345556, null),

('Имя26', 'Фамилия26', 'Отчество26', 89601112663, 2217, 123666, null),

('Имя27', 'Фамилия27', 'Отчество27', 89607712233, 2217, 773456, null),

('Имя28', 'Фамилия28', 'Отчество28', 89601112888, 2217, 123886, null),

('Имя29', 'Фамилия29', 'Отчество29', 89609912233, 2217, 993456, null),

('Имя30', 'Фамилия30', 'Отчество30', 89603312233, 2217, 333456, null)

go

select \* from Клиент

--

INSERT INTO Абонимент(ID\_Клиент, ID\_Вид\_абонимента, Дата\_получения)

VALUES (1, 18, '2021-09-03'),

(2, 17, '2021-04-03'),

(3, 15, '2022-11-03'),

(4, 20, '2022-09-13'),

(5, 14, '2022-09-03'),

(6, 13, '2022-09-03'),

(7, 9, '2022-09-03'),

(8, 5, '2022-09-03'),

(9, 10, '2022-09-03'),

(10, 16, '2022-10-03'),

(11, 15, '2022-09-03'),

(12, 19, '2022-09-03'),

(13, 21, '2020-09-03'),

(14, 20, '2022-09-03'),

(15, 11, '2022-09-03'),

(16, 12, '2021-11-03'),

(17, 6, '2022-06-03'),

(18, 5, '2022-09-03'),

(19, 10, '2022-07-11'),

(20, 16, '2022-09-03'),

(21, 15, '2022-09-03'),

(22, 1, '2022-09-03'),

(23, 12, '2012-02-03'),

(24, 15, '2019-09-22'),

(25, 15, '2023-08-03'),

(26, 15, '2022-09-03'),

(27, 14, '2022-09-03'),

(28, 5, '2022-09-03'),

(1, 13, '2022-09-03'),

(2, 12, '2022-09-03'),

(3, 5, '2022-09-03'),

(4, 7, '2022-09-03'),

(5, 16, '2022-09-03'),

(6, 11, '2022-09-03'),

(7, 19, '2022-09-03'),

(8, 15, '2022-09-03'),

(9, 11, '2022-09-03'),

(10, 14, '2022-09-03'),

(11, 15, '2022-09-03'),

(12, 5, '2022-09-03'),

(13, 1, '2022-09-03'),

(14, 11, '2022-09-03'),

(15, 17, '2022-09-03'),

(16, 18, '2022-09-03'),

(17, 19, '2022-09-03'),

(18, 12, '2022-09-03'),

(19, 21, '2022-09-03'),

(20, 20, '2022-09-03'),

(21, 11, '2022-09-03'),

(22, 11, '2022-09-03'),

(23, 11, '2022-09-03'),

(24, 15, '2022-09-03'),

(25, 14, '2022-09-03'),

(26, 13, '2022-09-03'),

(27, 13, '2022-09-03'),

(28, 9, '2022-09-03')

ALTER TABLE Тренер DROP COLUMN Количество\_инд\_тренеровок

go

INSERT INTO Тренер (ID\_Сотрудник)

VALUES (5),

(6),

(7),

(8)

go

INSERT INTO Расписание (ID\_Тренер, ID\_Группа, ID\_Зал, Дата, Время\_начала, Время\_конца)

VALUES (4, 1, 4, '2022-12-05', '13:00:00','14:00:00' ),

(4, 1, 4, '2022-12-07', '13:00:00','14:00:00' ),

(4, 1, 4, '2022-12-09', '13:00:00','14:00:00' ),

(4, 2, 4, '2022-12-05', '17:00:00','19:00:00' ),

(4, 2, 4, '2022-12-07', '17:00:00','19:00:00' ),

(4, 2, 4, '2022-12-09', '17:00:00','19:00:00' ),

(4, 3, 4, '2022-12-05', '20:00:00','22:00:00' ),

(4, 3, 4, '2022-12-07', '20:00:00','22:00:00' ),

(4, 3, 4, '2022-12-09', '20:00:00','22:00:00' ),

(4, 4, 3, '2022-12-06', '14:00:00','16:00:00' ),

(4, 4, 3, '2022-12-08', '14:00:00','16:00:00' ),

(4, 5, 3, '2022-12-06', '18:00:00','20:00:00' ),

(4, 5, 3, '2022-12-08', '18:00:00','20:00:00' ),

(1, 6, 5, '2022-12-10', '12:00:00','14:00:00' ),

(1, 7, 5, '2022-12-09', '20:00:00','22:00:00' ),

(2, 8, 1, '2022-12-06', '13:00:00','14:00:00' ),

(2, 8, 1, '2022-12-08', '13:00:00','14:00:00' ),

(2, 8, 1, '2022-12-10', '13:00:00','14:00:00' ),

(2, 9, 2, '2022-12-06', '18:00:00','19:30:00' ),

(2, 9, 2, '2022-12-08', '18:00:00','19:30:00' ),

(2, 9, 2, '2022-12-10', '18:00:00','19:30:00' )

go

INSERT INTO Абонименты\_групп (ID\_Абонимент, ID\_Группа )

VALUES ( 15, 1), ( 34, 1),( 37, 1), ( 42, 1),

( 49, 2), ( 51, 2), ( 50, 2),

(13 , 3), (47 , 3),

(16 , 4), (23 , 4), (30 , 4),

(14 , 5), (48 , 5),

(9 , 6),

(19 , 7)

INSERT INTO Индивидуальная\_тренеровка (ID\_Абонимент, ID\_Тренер, Дисциплина, Цена, Дата, Время\_начала, Время\_конца)

VALUES ( 1, 1, 'Фитнес', 1500, '2021-06-25', '20:00:00', '23:00:00'),

( 2, 2, 'Фитнес', 2500, '2021-06-15', '18:00:00', '20:00:00'),

( 3, 3, 'Растяжка', 1000, '2021-06-05', '21:00:00', '23:00:00'),

( 1, 1, 'Фитнес', 1500, '2021-06-30', '08:00:00', '11:00:00'),

( 2, 2, 'Фитнес', 2500, '2022-06-23', '22:00:00', '23:00:00'),

( 3, 3, 'Растяжка', 1000, '2021-06-03', '14:00:00', '16:00:00')

-- Запросы ----------------------------------------------------

go

-- вывод клиентов от фирм + названия и контактные телефоны фирм

select Имя, Фамилия, Название\_фирмы, Номер\_телефона from Клиент, Корпоротивный\_клиент

where Клиент.ID\_Корпоротивный\_клиент is not null and

Клиент.ID\_Корпоротивный\_клиент = Корпоротивный\_клиент.ID\_Корпоротивный\_клиент

-- Какиe Клиенты с групповыми абониментами зачислены в группы

select Имя, Фамилия, Название, Дисциплина, Группа.ID\_Группа from Клиент, Абонимент, Абонименты\_групп, Группа

where Клиент.ID\_Клиент = Абонимент.ID\_Клиент and

Абонимент.ID\_Абонимент = Абонименты\_групп.ID\_Абонимент and

Абонименты\_групп.ID\_Группа = Группа.ID\_Группа

-- Какие групповые абонименты есть у клиентов

select Клиент.Имя, Клиент.Фамилия, Вид\_абонимента.Название from Абонимент

full outer join Клиент on Абонимент.ID\_Клиент = Клиент.ID\_Клиент

full outer join Вид\_абонимента on Вид\_абонимента.ID\_Вид\_абонимента = Абонимент.ID\_Вид\_абонимента

left outer join Абонименты\_групп on Абонимент.ID\_Абонимент = Абонименты\_групп.ID\_Абонимент

where Абонимент.ID\_Вид\_абонимента in (10, 11,12, 16, 20, 21)

-- сортировка по дате получения абонимента ( по убыванию)

select Клиент.Имя, Клиент.Фамилия, Вид\_абонимента.Название, Абонимент.Дата\_получения from Абонимент

inner join Клиент on Абонимент.ID\_Клиент = Клиент.ID\_Клиент

inner join Вид\_абонимента on Вид\_абонимента.ID\_Вид\_абонимента = Абонимент.ID\_Вид\_абонимента

order by Абонимент.Дата\_получения desc

-- Виды абониментов отсортированные по цене на увеличение

select Название, Цена , Конец\_срока as Длительность, Время\_начала as [Время начала], Время\_конца as [Время конца] from Вид\_абонимента

order by Цена asc

-- Какую цену заплатили корпоративные клиенты за все время в 2022 году

select Корпоротивный\_клиент.Название\_фирмы, sum(Вид\_абонимента.Цена) as [Сумма за все время] from Абонимент

inner join Клиент on Абонимент.ID\_Клиент = Клиент.ID\_Клиент

inner join Корпоротивный\_клиент on Клиент.ID\_Корпоротивный\_клиент = Корпоротивный\_клиент.ID\_Корпоротивный\_клиент

inner join Вид\_абонимента on Абонимент.ID\_Вид\_абонимента = Вид\_абонимента.ID\_Вид\_абонимента

where Абонимент.Дата\_получения > '2022-01-01'

group by Корпоротивный\_клиент.Название\_фирмы

--Отсортированный список с суммами цен индивидуальных занятий тренеров

select Тренер.ID\_Тренер, Сотрудник.Имя, Сотрудник.Фамилия , sum(Индивидуальная\_тренеровка.Цена) as [Сумма за все время] from Тренер

inner join Индивидуальная\_тренеровка on Тренер.ID\_Тренер = Индивидуальная\_тренеровка.ID\_Тренер

inner join Сотрудник on Тренер.ID\_Сотрудник = Сотрудник.ID\_Сотрудник

group by Сотрудник.Имя, Сотрудник.Фамилия, Тренер.ID\_Тренер

order by Сотрудник.Имя, Сотрудник.Фамилия

-- количество клиентов в фитнес клубе и сколько они имеют абониментов

select Клиент.ID\_Клиент, count(Абонимент.ID\_Клиент) as [Количество абониментов] from Клиент

full join Абонимент on Клиент.ID\_Клиент = Абонимент.ID\_Клиент

group by Абонимент.ID\_Клиент, Клиент.ID\_Клиент

-- Поиск ID клиента по имени и фамилии

select Клиент.ID\_Клиент, Клиент.Имя, Клиент.Фамилия, Клиент.Телефон from Клиент

where Имя = 'Имя21' and Фамилия = 'Фамилия21'

--Изменение Номера телефона

update Клиент set Телефон ='89601889925'

where ID\_Клиент = 30

--Удалить последнюю строку из таблицы

delete from Индивидуальная\_тренеровка where ID\_Индивидуальная\_тренировка=(select max(ID\_Индивидуальная\_тренеровка) from Индивидуальная\_тренеровка)

# **Заключение**

В результате проделанной работы решены и сделаны следующие задачи:

1. Выявлены основные понятия.

2. Была описана выбранная предметную область , процессы и объекты, которые были сохранены в нашей БД.

3. Была изображена схема данных на логическом и физическом уровнях проектирования и доказано, что все таблицы удовлетворяют 3-ей нормальной форме.

4. Была показана работа всех видов запросов, объединений, подзапросов, производных таблиц, автовычисляемых полей.

Таким образом, после работы с созданием БД «Фитнес-клуб», можно прийти к выводу, что использование СУБД даёт много возможностей пользователям. Мы получаем возможность не только структурировано хранить данные, но и без проблем манипулировать данными