МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«Санкт-Петербургский национальный исследовательский университет  
информационных технологий, механики и оптики»

Факультет информационных технологий и программирования

Кафедра информационных систем

Проект базы данных

Выполнила студентка группы № M3200  
Авдеева Ульяна Константиновна

Проверил:  
Повышев Владислав Вячеславович

САНКТ -ПЕТЕРБУРГ   
2017

Описание предметной области

Предметом изучения является модель базы данных университета с упором на расписание занятий секций спортивного клуба и расписания учебных занятий. Обучаясь, студент желает посещать и то, и то. База являет собой совокупность всех данных, необходимых для координации процесса.

Сущности

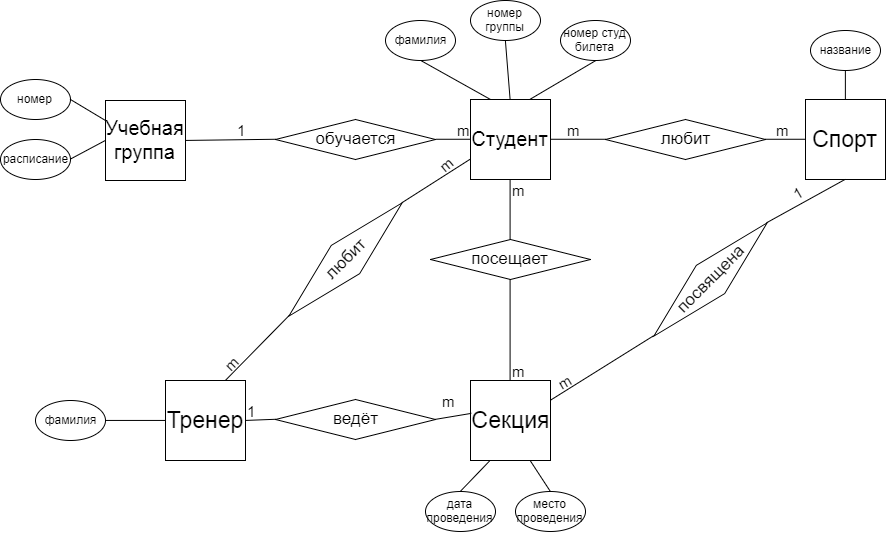
1. Студент. Обучается в учебной группе, имеет своё личное расписание, посещает секции, любит/не любит какие-то виды спорта и каких-то преподавателей. Обучается в одной учебной группе, может любить несколько видов спорта, несколько тренеров. Может посещать много секций.
2. Учебная группа. В учебной группе обучается студент. У каждой группы есть своё расписание. В одной учебной группе может быть много студентов, но один студент может быть только в одной группе.
3. Секция. Секции посещают студенты. Их проводят тренеры. Одна секция может соответствовать лишь одному виду спорта. Секцию ведет строго один тренер.
4. Тренер. Он ведёт занятия в секциях, его может любить/не любить конкретный студент. Ведёт определённые виды спорта. Один тренер может вести много видов спорта, много секций.
5. Вид спорта. Разные виды спорта бывают в разных секциях. Студент может позитивно или негативно относиться к виду спорта.

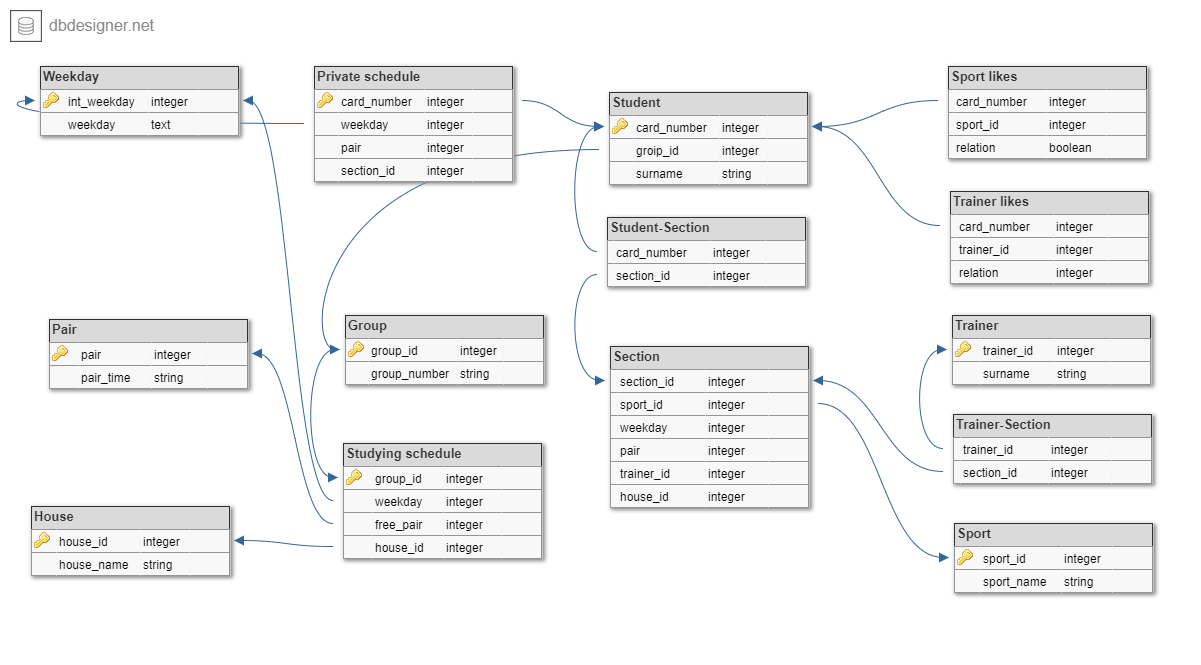
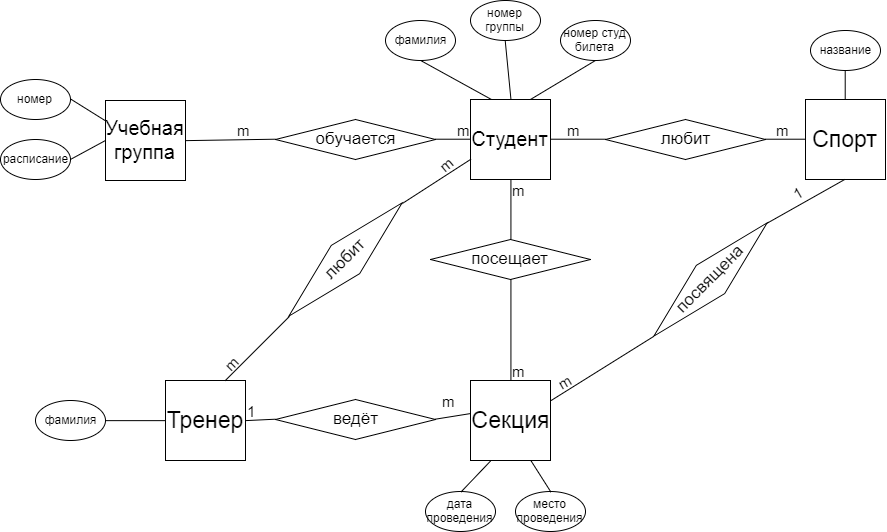
Функционал

В ходе выполнения работы были достигнуты все поставленные цели. Основной из таковых была реализация функции поиска потенциальных секций для конкретного студента с учётом его расписания (включая местоположение корпуса). Все процедуры, функции, представления, триггеры и курсор представлены ниже. В целом база готова для использования в полноценном полезном приложении, реализующим следующие функции:

* Запись в секцию
* Перенос/отмена занятия
* Наиболее интересные предложения по секциям
* Система оповещения студентов при переносе занятия
* Проследить активность тренера
* Статистика по преподавателям/видам спорта
* Вывод расписания работы секций

Нотация Чена, реляционная модель





Скрипт создания базы

CREATE TABLE "Private schedule" (

"card\_number" INT,

"weekday" INT,

"pair" INT,

"section\_id" INT,

PRIMARY KEY ("card\_number"),

SET IDentity\_Insert off

)

CREATE TABLE "Weekday" (

"int\_weekday" INT,

"weekday" TEXT,

PRIMARY KEY ("int\_weekday"),

SET IDentity\_Insert off

)

CREATE TABLE "Student" (

"card\_number" INT,

"groip\_id" INT,

"surname" VARCHAR(255),

PRIMARY KEY ("card\_number"),

SET IDentity\_Insert off

)

CREATE TABLE "Student-Section" (

"card\_number" INT,

"section\_id" INT,

SET IDentity\_Insert off

)

CREATE TABLE "Group" (

"group\_id" INT,

"group\_number" VARCHAR(255),

PRIMARY KEY ("group\_id"),

SET IDentity\_Insert off

)

CREATE TABLE "Studying schedule" (

"group\_id" INT,

"weekday" INT,

"free\_pair" INT,

"house\_id" INT,

PRIMARY KEY ("group\_id"),

SET IDentity\_Insert off

)

CREATE TABLE "Pair" (

"pair" INT ,

"pair\_time" VARCHAR(255),

PRIMARY KEY ("pair"),

SET IDentity\_Insert off

)

CREATE TABLE "Section" (

"section\_id" INT,

"sport\_id" INT,

"weekday" INT,

"pair" INT,

"trainer\_id" INT,

"house\_id" INT,

SET IDentity\_Insert off

)

CREATE TABLE "House" (

"house\_id" INT,

"house\_name" VARCHAR(255),

PRIMARY KEY ("house\_id"),

SET IDentity\_Insert off

)

CREATE TABLE "Sport likes" (

"card\_number" INT,

"sport\_id" INT,

"relation" BOOLEAN,

SET IDentity\_Insert off

)

CREATE TABLE "Trainer likes" (

"card\_number" INT,

"trainer\_id" INT,

"relation" INT,

SET IDentity\_Insert off

)

CREATE TABLE "Trainer" (

"trainer\_id" INT,

"surname" VARCHAR(255),

PRIMARY KEY ("trainer\_id")

)

CREATE TABLE "Trainer-Section" (

"trainer\_id" INT,

"section\_id" INT,

SET IDentity\_Insert off

)

CREATE TABLE "Sport" (

"sport\_id" INT,

"sport\_name" VARCHAR(255),

PRIMARY KEY ("sport\_id"),

SET IDentity\_Insert off

)

ALTER TABLE "Private schedule" ADD CONSTRAINT "Private schedule\_fk0" FOREIGN KEY ("card\_number") REFERENCES "Student"("card\_number")

ON DELETE CASCADE

ALTER TABLE "Private schedule" ADD CONSTRAINT "Private schedule\_fk1" FOREIGN KEY ("weekday") REFERENCES "Weekday"("int\_weekday")

ON DELETE CASCADE

ALTER TABLE "Student" ADD CONSTRAINT "Student\_fk0" FOREIGN KEY ("groip\_id") REFERENCES "Group"("group\_id")

ON DELETE CASCADE

ALTER TABLE "Student-Section" ADD CONSTRAINT "Student-Section\_fk0" FOREIGN KEY ("card\_number") REFERENCES "Student"("card\_number")

ON DELETE CASCADE

ALTER TABLE "Student-Section" ADD CONSTRAINT "Student-Section\_fk1" FOREIGN KEY ("section\_id") REFERENCES "Section"("section\_id")

ON DELETE CASCADE

ALTER TABLE "Group" ADD CONSTRAINT "Group\_fk0" FOREIGN KEY ("group\_id") REFERENCES "Studying schedule"("group\_id")

ON DELETE CASCADE

ALTER TABLE "Studying schedule" ADD CONSTRAINT "Studying schedule\_fk0" FOREIGN KEY ("weekday") REFERENCES "Weekday"("int\_weekday")

ON DELETE CASCADE

ALTER TABLE "Studying schedule" ADD CONSTRAINT "Studying schedule\_fk1" FOREIGN KEY ("free\_pair") REFERENCES "Pair"("pair")

ON DELETE CASCADE

ALTER TABLE "Studying schedule" ADD CONSTRAINT "Studying schedule\_fk2" FOREIGN KEY ("house\_id") REFERENCES "House"("house\_id")

ON DELETE CASCADE

ALTER TABLE "Section" ADD CONSTRAINT "Section\_fk0" FOREIGN KEY ("sport\_id") REFERENCES "Sport"("sport\_id")

ON DELETE CASCADE

ALTER TABLE "Sport likes" ADD CONSTRAINT "Sport likes\_fk0" FOREIGN KEY ("card\_number") REFERENCES "Student"("card\_number")

ON DELETE CASCADE

ALTER TABLE "Trainer likes" ADD CONSTRAINT "Trainer likes\_fk0" FOREIGN KEY ("card\_number") REFERENCES "Student"("card\_number")

ON DELETE CASCADE

ALTER TABLE "Trainer-Section" ADD CONSTRAINT "Trainer-Section\_fk0" FOREIGN KEY ("trainer\_id") REFERENCES "Trainer"("trainer\_id")

ON DELETE CASCADE

ALTER TABLE "Trainer-Section" ADD CONSTRAINT "Trainer-Section\_fk1" FOREIGN KEY ("section\_id") REFERENCES "Section"("section\_id")

ON DELETE CASCADE

Триггеры, процедуры, функции, представления, курсор

Процедуры:

1. Записать студента в секцию, (сделать пометки в его личном расписании, записать в таблицу секций)

CREATE PROCEDURE Procedure\_enroll @stud int, @sectid int

AS

INSERT INTO [Student-Section]

VALUES (@stud, @sectid)

DECLARE @weekday INT

SELECT @weekday = MAX([Section].weekday) FROM [Section] WHERE [Section].section\_id=@sectid

DECLARE @pair INT

SELECT @pair = MAX([Section].pair) FROM [Section] WHERE [Section].section\_id=@sectid

INSERT INTO [Private schedule]

VALUES (@stud, @weekday, @pair, @sectid)

GO

1. Перенести секцию на другое время (поменять везде день недели, номер пары)

CREATE PROCEDURE Procedure\_reschedule @sectid INT, @weekday INT, @pair int

AS

UPDATE [Section]

SET [Section].weekday=@weekday

WHERE [Section].section\_id=@sectid

UPDATE [Section]

SET [Section].pair=@pair

WHERE [Section].section\_id=@sectid

UPDATE [Private schedule]

SET [Private schedule].weekday=@weekday

WHERE [Private schedule].section\_id=@sectid

UPDATE [Private schedule]

SET [Private schedule].pair=@pair

WHERE [Private schedule].section\_id=@sectid

GO

1. Уволить тренера и всю информацию о нём ( его секции, лайк/дизлайк и т.д.)

CREATE PROCEDURE Lay\_off @train int

AS

DELETE FROM [Student-Section] WHERE section\_id in (SELECT section\_id FROM Section WHERE trainer\_id=@train)

DELETE FROM [Private schedule] WHERE section\_id in (SELECT section\_id FROM Section WHERE trainer\_id=@train)

DELETE FROM Section WHERE trainer\_id=@train

DELETE FROM Trainer WHERE trainer\_id=@train

DELETE FROM [Trainer like] WHERE trainer\_id=@train

DELETE FROM [Trainer-Section] WHERE trainer\_id=@train

GO

Курсор:

Вывести все секции, которые может посещать студент согласно его учебному расписанию

ALTER PROCEDURE Select\_section\_to\_me @cardnum int

AS

SET NOCOUNT ON;

DECLARE @group INT;

SELECT @group = [Student].groip\_id FROM [Student] WHERE [Student].card\_number=@cardnum

DECLARE @sect\_id INT, @sport\_id INT, @weekday INT, @pair INT, @house\_id INT;

DECLARE section\_cursor CURSOR FOR

SELECT [Section].section\_id, [Section].sport\_id, [Section].weekday, [Section].pair, [Section].house\_id FROM [Section]

;

OPEN section\_cursor

FETCH NEXT FROM section\_cursor

INTO @sect\_id, @sport\_id, @weekday, @pair, @house\_id

WHILE @@FETCH\_STATUS = 0

BEGIN

DECLARE @ok INT

SELECT @ok = COUNT(\*) FROM [Studying schedule] WHERE ([Studying schedule].group\_id=@group AND [Studying schedule].weekday = @weekday AND [Studying schedule].free\_pair = @pair AND ([Studying schedule].house\_id = @house\_id OR [Studying schedule].house\_id = -1));

IF (@ok = 1)

SELECT [Weekday].weekday, [Pair].pair\_time, [House].house\_name, [Sport].sport\_name, [Trainer].surname

FROM [Weekday]

JOIN [Section]

ON [Weekday].int\_weekday=[Section].weekday

JOIN [Sport]

ON [Sport].sport\_id=[Section].sport\_id

JOIN [Trainer]

ON [Trainer].trainer\_id=[Section].trainer\_id

JOIN [Pair]

ON [Pair].pair=@pair

JOIN [House]

ON [House].house\_id=[Section].house\_id

WHERE @sect\_id=[Section].section\_id AND [Section].weekday=@weekday AND [Pair].pair=@pair

FETCH NEXT FROM section\_cursor

INTO @sect\_id, @sport\_id, @weekday, @pair, @house\_id

END

CLOSE section\_cursor;

DEALLOCATE section\_cursor;

GO

Триггеры:

1. При обновлении времени проведения занятия вывести фамилии тех, кого надо уведомить

ALTER TRIGGER notify

ON [Section]

AFTER UPDATE

AS

BEGIN

SELECT Student.surname

FROM Student

JOIN [Student-Section]

ON Student.card\_number=[Student-Section].card\_number

JOIN [INSERTED]

ON [INSERTED].section\_id=[Student-Section].section\_id

END

GO

1. При вводе фамилии тренера привести его к нормальному виду (пЕтЯ -> петя)

ALTER TRIGGER Paste\_trainer

ON [Trainer]

AFTER INSERT

AS

BEGIN

DECLARE @st nvarchar(250)

SELECT TOP 1 @st = [INSERTED].surname FROM [INSERTED]

UPDATE [Trainer]

SET [Trainer].surname = LOWER (@st)

WHERE [Trainer].trainer\_id in (SELECT trainer\_id FROM [INSERTED]);

END

GO

1. При попытке удалить корпус не дать удалить корпус

CREATE TRIGGER Delete\_house

ON [House]

INSTEAD OF DELETE

AS

BEGIN

PRINT N'You can not do this'

END

GO

Функции:

1. Вернуть фамилии участников секции, день недели, пару и вид спорта по id секции

ALTER FUNCTION Select\_section (@sect int) RETURNS TABLE AS

RETURN (

SELECT [Student-Section].section\_id,[Student].surname, [Sport].sport\_name, [Weekday].weekday, [Pair].pair\_time

FROM [Student]

JOIN [Student-Section]

ON [Student].card\_number=[Student-Section].card\_number

JOIN [Section]

ON [Section].section\_id=[Student-Section].section\_id

JOIN [Sport]

ON [Sport].sport\_id=[Section].sport\_id

JOIN [Weekday]

ON [Weekday].int\_weekday=[Section].weekday

JOIN [Pair]

ON [Pair].pair=[Section].pair

WHERE [Student-Section].section\_id=@sect

);

GO

1. Вывести все секции, которые ведёт данный тренер: фамилия, день ндели, пара, название спорта

ALTER FUNCTION All\_trainer\_activity (@trainid int) RETURNS TABLE AS

RETURN (

SELECT [Section].section\_id, [Trainer].surname, [Weekday].weekday, [Pair].pair, [Sport].sport\_name

FROM [Trainer]

JOIN [Section]

ON [Trainer].trainer\_id=[Section].trainer\_id

JOIN [Weekday]

ON [Weekday].int\_weekday=[Section].weekday

JOIN [Pair]

ON [Pair].pair=[Section].pair

JOIN [Sport]

ON [Sport].sport\_id=[Section].sport\_id

WHERE [Section].trainer\_id=@trainid

);

GO

1. Вывести фамилии студентов, которые любят данного преподавателя

CREATE FUNCTION lovely\_students (@trainerid int) RETURNS TABLE AS

RETURN (

SELECT Student.surname

FROM Student

JOIN [Trainer likes]

ON Student.card\_number=[Trainer likes].card\_number

WHERE ([Trainer likes].trainer\_id=@trainerid AND [Trainer likes].relation=1))

GO

Представления:

1. Вывести все секции - зал, название спорта, фамилия, день недели, пара

ALTER VIEW view\_gyms

AS SELECT [Section].section\_id, [House].house\_name, [Sport].sport\_name, [Trainer].surname, [Weekday].weekday, [Pair].pair\_time

FROM Section

JOIN House

ON Section.house\_id=House.house\_id

JOIN Sport

ON Sport.sport\_id=Section.sport\_id

JOIN Trainer

ON Trainer.trainer\_id=Section.trainer\_id

JOIN [Weekday]

ON [Weekday].int\_weekday=Section.weekday

JOIn Pair

ON Pair.pair=Section.pair

GO

1. Студент / секция / день недели / пара

CREATE VIEW view\_students

AS SELECT [Student].surname, [Sport].sport\_name, [Weekday].weekday, [Pair].pair\_time

FROM [Student-Section]

JOIN [Section]

ON [Student-Section].section\_id=Section.section\_id

JOIN Student

ON Student.card\_number=[Student-Section].card\_number

JOIN [Weekday]

ON [Weekday].int\_weekday=Section.weekday

JOIN [Sport]

ON Sport.sport\_id=Section.sport\_id

JOIN [Pair]

ON Pair.pair=Section.pair

GO

1. Все студенты с их нелюбимыми видами спорта

ALTER VIEW view\_bad\_sports

AS SELECT Sport.sport\_name,Student.surname

FROM Sport

JOIN [Sport likes]

ON Sport.sport\_id=[Sport likes].sport\_id

JOIN [Student]

ON Student.card\_number=[Sport likes].card\_number

WHERE [Sport likes].relation=0

GO