**Приложение «StudtoDo» – планировщик задач для студента.**

# Описание ПО.

Некоторый студент имеет набор дел. Дела разделяются по области деятельности: на учебную и внеучебную. Рассматривая учебные планы, студент имеет представление обо всех предметах и о их неполных характеристиках с начала семестра. Полными характеристики могут стать в течение семестра.

Название предмета, ФИО преподавателя и его описание – это то, что студент точно будет знать с начала семестра, а даты сдачи «заданий», категория, к которой относится преподаватель: «допускающий сдачу заданий невовремя» и «не допускающий сдачу заданий невовремя», категория сложности заданий: «сложно», «средне», «просто», – это то, что студент будет узнавать в течение семестра. К «заданиям» относятся лабораторные, сдача документов, экзамены, контрольные и другие контрольно-измерительных мероприятия.

Также студент обладает определенной базой знаний, составляющие которой могут относиться либо к «заданиям» учебной деятельности, либо к делам или событиям внеучебной деятельности.

К учебной деятельности относится множество уникальных предметов, к каждому из которых относиться некая информация, описывающая предмет: название предмета, ФИО преподавателя, его почта и/или номер телефона, категория преподавателя, описание, набор уникальных заданий, к каждому из которых относиться некая совокупность информации, описывающая задание: название задания или его краткое описание, категория задания, описание, дата сдачи.

К внеучебной деятельности относится множество дел и событий, к каждому из которых относится некая информация, описывающая дело: название дела или краткое описание, дата завершения, описание дела.

# Концептуальная модель.

На основе анализа предметной области «планировщик задач для студента», были выведены следующие информационные объекты, которые необходимо хранить в базе данных:

ОБЛАСТЬ\_ДЕЯТЕЛЬНОСТИ, ПРЕДМЕТ, ЗАДАНИЕ, КАТЕГОРИЯ\_ЗАДАНИЯ, ПРЕПОДАВАТЕЛЬ, КАТЕГОРИЯ\_ПРЕПОДАВАТЕЛЯ, ОБЛАСТЬ\_ЗНАНИЙ.

Каждый из выделенных информационных объектов имеет следующие атрибуты:

ОБЛАСТЬ\_ДЕЯТЕЛЬНОСТИ – название области деятельности;

ПРЕДМЕТ – название предмета, описание предмета;

ПРЕПОДАВАТЕЛЬ – фамилия, имя и отчество, почта, номер телефона;

КАТЕГОРИЯ\_ПРЕПОДАВАТЕЛЯ – название категории преподавателя;

ЗАДАНИЕ – краткая сводка о задании, описание задания, дата сдачи;

КАТЕГОРИЯ\_ЗАДАНИЯ – название категории задания;

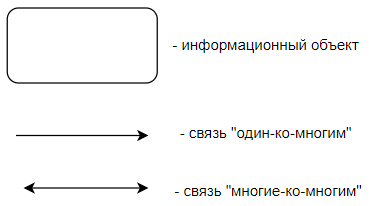
ОБЛАСТЬ\_ЗНАНИЙ – название, большой текст (информация, помогающая с заданием);

На рисунке 1 приведена концептуальная модель предметной области «планировщик заданий для студента» в виде ER-диаграммы.



Рис. 1. Концептуальная схема базы данных «планировщик заданий для студента».

Для этой диаграммы приняты следующие обозначения:



# Логическая модель.

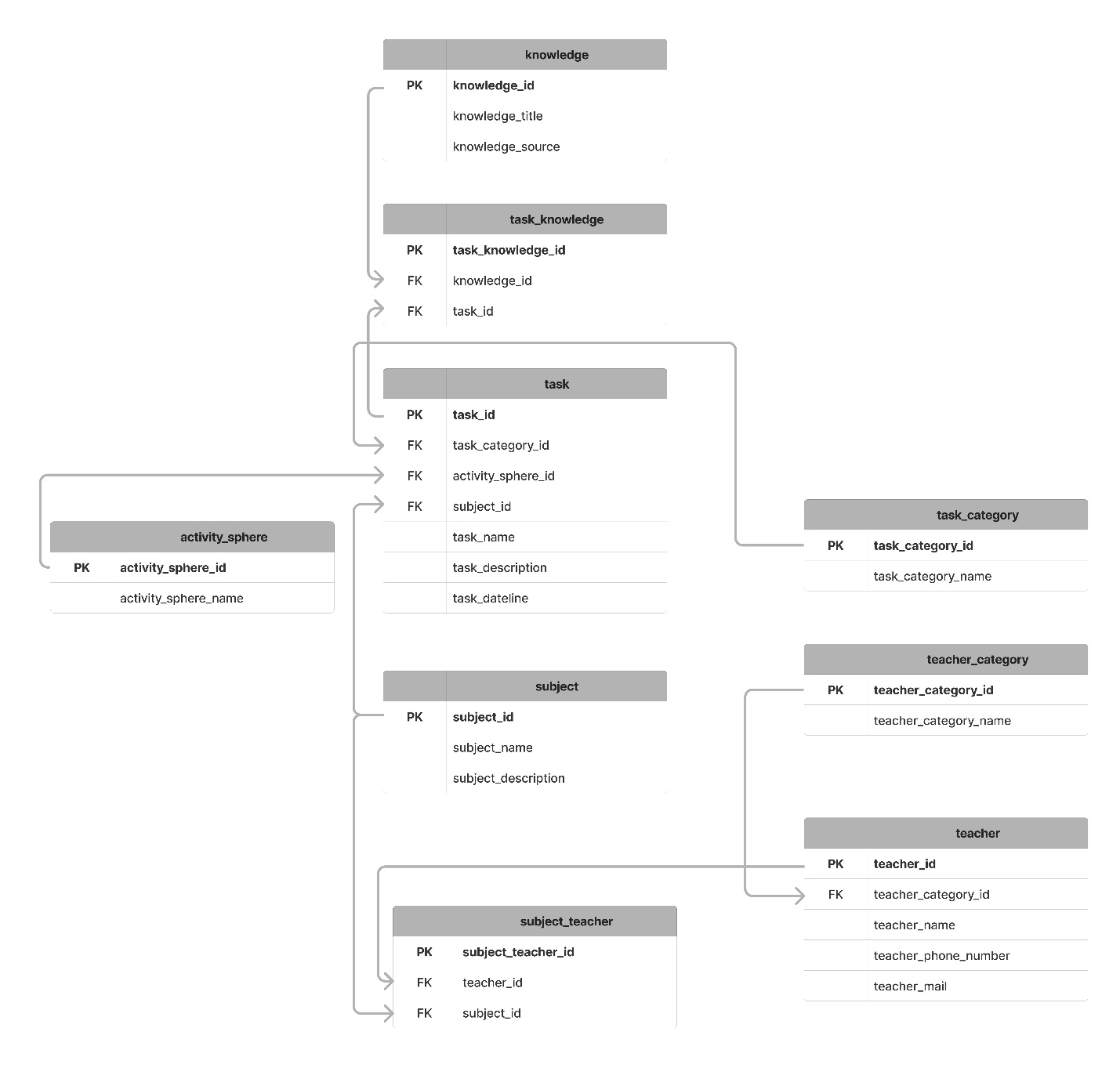
На рисунке 2 приведена логическая модель базы данных «Планировщик задач для студента». 

Рис. 2. Логическая схема базы данных «Планировщик задача для студента».

# Физическая модель.

В результате проектирования базы данных были получены информационные объекты, реализованные в виде реляционных таблиц. Ниже приведены SQL-запросы для их создания, в Приложении А приведено наполнение таблиц базы данных.

        create table if not exists knowledge (

            knowledge\_id integer primary key autoincrement,

            knowledge\_title varchar(100),

            knowledge\_source text

        );

        create table if not exists activity\_sphere(

            activity\_sphere\_id integer primary key autoincrement,

            activity\_sphere\_name varchar(30)

        );

        create table if not exists task\_category(

            task\_category\_id integer primary key autoincrement,

            task\_category\_name varchar(30)

        );

        create table if not exists subject(

            subject\_id integer primary key autoincrement,

            subject\_name varchar(30),

            subject\_description text

        );

        create table if not exists task (

            task\_id integer primary key autoincrement,

            task\_category\_id int,

            activity\_sphere\_id int,

            subject\_id int,

            task\_name varchar(100),

            task\_description text,

            task\_dateline date,

            foreign key (task\_category\_id) references task\_category (task\_category\_id) on delete cascade,

            foreign key (activity\_sphere\_id) references activity\_sphere (activity\_sphere\_id) on delete cascade,

            foreign key (subject\_id) references subject (subject\_id) on delete cascade

        );

        create table if not exists task\_knowledge (

            task\_knowledge\_id integer primary key autoincrement,

            knowledge\_id int,

            task\_id int,

            foreign key (knowledge\_id) references knowledge (knowledge\_id) on delete cascade,

            foreign key (task\_id) references task (task\_id) on delete cascade

        );

        create table if not exists teacher\_category (

            teacher\_category\_id integer primary key autoincrement,

            teacher\_category\_name

        );

        create table if not exists teacher (

            teacher\_id integer primary key autoincrement,

            teacher\_category\_id int,

            teacher\_name varchar(35),

            teacher\_phone\_number varchar(11),

            teacher\_mail varchar(50),

            foreign key (teacher\_category\_id) references teacher\_category (teacher\_category\_id) on delete cascade

        );

        create table if not exists subject\_teacher (

            subject\_teacher\_id integer primary key autoincrement,

            teacher\_id int,

            subject\_id int,

            foreign key (teacher\_id) references teacher (teacher\_id) on delete cascade,

            foreign key (subject\_id) references subject (subject\_id) on delete cascade

        );

## ПРИЛОЖЕНИЕ А. Наполнение таблиц базы данных.

        insert into activity\_sphere (activity\_sphere\_name) values

        ("Учебная"),

        ("Внеучебная");

        insert into task\_category (task\_category\_name) values

        ("Сложно"),

        ("Средне"),

        ("Просто");

        insert into subject (subject\_name, subject\_description) values

        ("ПП", "Параллельное программирование"),

        ("СиИТ", "Сетевые и интернет технологии");

        insert into task (task\_category\_id, activity\_sphere\_id, subject\_id, task\_name, task\_description, task\_dateline) values

        (2, 1, 1, "lab\_1", "lab\_1 description", date()),

        (2, 2, null, "Сходить за хлебом", "Сходить за белым резанным хлебом", date()),

        (3, 1, 2, "lab\_1", "lab\_1 description", "2024-02-12");

        insert into knowledge (knowledge\_title, knowledge\_source) values

        ("css", "что-то о css"),

        ("html", "что-то о html"),

        ("ООП", "что-то об ООП"),

        ("c++", "что-то о c++"),

        ("Структуры данных", "что-то о структурах данных");

        insert into task\_knowledge (knowledge\_id, task\_id) values

        (1, 3),

        (5, 3),

        (5, 1),

        (3, 1);

        insert into teacher\_category (teacher\_category\_name) values

        ("Допускающий сдачу заданий невовремя"),

        ("Не допускающий сдачу заданий невовремя");

        insert into teacher (teacher\_category\_id, teacher\_name, teacher\_phone\_number, teacher\_mail) values

        (1, "Озерова Галина Павловна", 89999999999, null),

        (2, "Чусов Андрей Александрович", null, null);

        insert into subject\_teacher (teacher\_id, subject\_id) values

        (1, 2),

        (2, 1);