

Тестирование

Вам поручено реализовать новый функционал системы. База данных и код в вашем полном распоряжении.

У компании заключены договора с Клиентами на консалтинговые услуги. В рамках договора клиенты могут присылать в компанию запросы по разным каналам(email, апи, телефон), как новые, так и уточняющие другой запрос.

Свойства модели Клиент: название, ключевитость(не ключевой, ключевой, суперключевой).

Свойства модели Запрос: тип(консультация, заказ документов), дата создания, клиент, содержание, ответ, канал(email, апи, телефон), приоритет, ссылка на модель запроса по которому прислано уточнение.

Для оптимизации работы сотрудников в компании вводится система приоритезации запросов.

При создании запроса необходимо установить приоритет, определив его вес по набору критериев. Общий приоритет запроса вычисляется исходя из суммы весов каждого критерия, которому соответствует запрос. Количество и вычисление критериев, как и вес могут меняться. Критерии определяются разработчиками, в каждом критерии своя логика определения подходит или нет критерий запросу.

Весы критериев могут меняться, в том числе пользователями(сотрудникам) и связаны с типом запроса.

Для примера как может выглядеть таблица весов и соответствующих им критериев по типам задач

Название критерия	Вес для типа консультация	Вес для типа заказ документов
ключевой клиент	2	3
суперключевой клиент	3	4
первый запрос от клиента	2	2
наличие уточнений от клиента	4	1

Ваша задача определить архитектуру приложения для решения поставленной задачи:

1. Придумайте и схематично нарисуйте схему базы данных, по Вашему мнению, достаточную для выполнения данной задачи
2. Придумайте и схематично изобразите структуру классов и методов(без реализации) для получения запроса по каналам email, апи(запрос из внешней системы), телефон(заполнение формы на сайте)
3. Придумайте и схематично изобразите структуру классов и методов(без реализации) для сервиса определения приоритета запроса по критериям и весам и сервиса создания запроса с учетом многообразия каналов.

4. Напишите SQL запрос, который вернет клиентов, которые хотя бы раз присылали запрос, но не присылали больше одного уточнения к запросу.