# er-Диаграмма

Поставлена задача: **разработать модель данных для программного комплекса управления рассылкой электронных сообщений.**

Оператор передает информацию о сообщении в пользовательской части и при помощи API передает его на сервер, где информация о сообщении формализуется, заносится в базу данных и передается сервисам доставки.

В результате анализа предметной области можно обозначить несколько сущностей:

Поисковые службы занимаются поиском пропавших объектов, с использованием компьютера и мобильных телефонов. Компьютером пользуется оператор, на котором он назначает задачи для участников поисковой группы. Мобильным приложением пользуются участники поисковой группы.

В результате анализа процесс обучения можно обозначить несколько сущностей. Вот описание некоторых основных сущностей:

* **Messages**
  + id (int) – суррогатный первичный ключ;
  + userId (int) – id отправителя;
  + destinationDate (dateTime) – дата и время успешной доставки;
  + theme (nvarchar(100)) – тема сообщения;
  + body (text) – тело сообщения;
  + size (int) – размер сообщения в единицах (байт);
  + isScheduled (int) – логическое поле – запланирована ли отправка сообщения;
  + scheduleDate (dateTime) – запланированная дата и время, когда необходимо отправить сообщение;
  + isSent (int) – логическое поле – успешно ли доставлено сообщение;
  + journalId (int) – id журнала, в котором содержится данное сообщение;
  + queueId (int) – id очереди, в которой содержится данное сообщение.
* **Users**
  + id (int) – суррогатный первичный ключ;
  + userName (nvarchar(100)) – имя пользователя;
  + email (nvarchar(100)) – почтовый адрес пользователя;
  + passwordHash (nvarchar(100)) – хэш, полученный из пароля пользователя;
  + isActive (int) – логическое поле – статус доступа пользователя;
  + journalId (int) – Id журнала данного пользователя;
  + queueId (int) – id очереди, в которую помещаются сообщения данного пользователя.
* **DeliveryServices**
  + id (int) – суррогатный первичный ключ;
  + serviceName (nvarchar(100)) – название сервиса доставки;
  + standartPriority (int) – значение приоритета данного сервиса доставки при переборе списка сервисов во время отправки по умолчанию.
* **Accesses**
  + id (int) – суррогатный первичный ключ;
  + startDate (dateTime) – дата и время начала подписки пользователя;
  + endDate (dateTime) – дата и время конца подписки пользователя;
  + tariffId (int) – id тарифа пользователя.
* **Tariffs**
  + id (int) – суррогатный первичный ключ;
  + tariffName (nvarchar(255)) – название тарифа;
  + cost (int) – стоимость тарифа.
* **SchedulePeriods**
  + ID (int) – суррогатный первичный ключ;
  + startDate (dateTime) – дата и время начала периода, когда необходимо доставить сообщение;
  + endDate (dateTime) – дата и время конца периода, когда необходимо доставить сообщение.
* **Files**
  + id (int) – суррогатный первичный ключ;
  + line (nvarchar(127)) – ссылка на файл, хранящийся на сервере;
  + size (int) – размер файла, на который указывает ссылка в поле link;
  + typeId – тип файла, на который указывает ссылка в поле link.
* **DeliveryStatuses**
  + id (int) – суррогатный первичный ключ;
  + name (nvarchar(127)) – статус доставки.
* **Roles**
  + id (int) – суррогатный первичный ключ;
  + roleName (nvarchar(127)) – роль (администратор или оператор).
* **Sales**
  + id (int) – суррогатный первичный ключ;
  + name (nvarchar(255)) – название скидки на тариф;
  + value (int) – размер скидки на тариф.

ER диаграмма представлена на рис. 1.

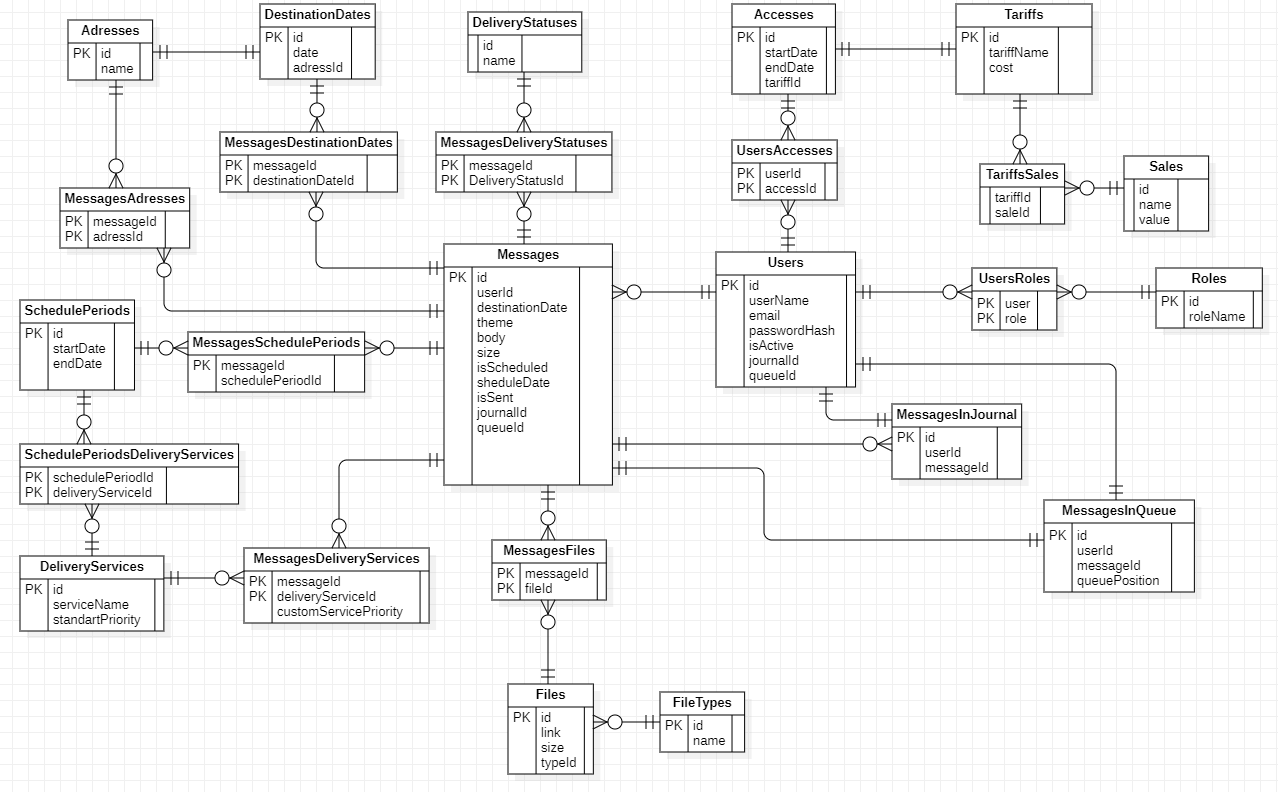


Рис. 1. ER диаграмма

# DF-диаграмма

Поставлена задача: разработать модель потоков данных **для программного комплекса управления рассылкой электронных сообщений**

На контекстной диаграмме (рис. 2) определены базовый блок «Система рассылки и учета электронных сообщений» и внешняя сущность «Пользователь», хранилища данных «Журнал», «Очередь», «Письма», а также информационные потоки между ними.

На диаграмме первого уровня (рис. 3) выполнена декомпозиция базового процесса на три подпроцесса. На диаграмме второго уровня проведена детализация процесса «Инициировать отправку письма» (рис. 4). На диаграммах третьего уровня проведена детализация процесса «Отправить письмо» (рис. 5).



Рис. 2. Контекстная диаграмма



Рис. 3. Диаграмма первого уровня модели потоков данных



Рис. 4. Диаграмма второго уровня модели потоков данных. Процесс «Инициировать отправку письма»



Рис. 5. Диаграмма третьего уровня потоков данных. Процесс «Отправить письмо»

# IDEF0 диаграмма

Поставлена задача: *отправить сообщение.* Необходимо определить функциональные требования к программной системе.

На контекстной диаграмме (рис. 6) определен функциональный блок «Отправить сообщение через систему», механизм, предписание, входные и выходные данные.

На диаграмме первого уровня (рис. 7) выполнена декомпозиция базового блока «Отправить сообщение через систему». Этот блок содержит четыре подфункции: «Сформировать письмо», «Инициировать отправку сообщения», «Занести информацию о доставке в журнал». Проведена детализация функций «Сформировать письмо» (рис. 8) и «Инициировать отправку сообщения» (рис. 9)



Рис. 7. Контекстная диаграмма функциональной модели «Поиск объектов»



Рис. 8. Диаграмма первого уровня функциональной модели



Рис. 9. Диаграмма второго уровня для блока «Сформировать письмо»



Рис. 10. Диаграмма второго уровня для блока «Инициировать отправку сообщения»