

## Пояснительная записка

Аверичкина Виктория  
БПИ184

### 1) Кому нужна разработка

Разработка для телемаркетинговой компании, которая подбирает компаниям клиентов, а клиентам компании по их нуждам и предложению.

Для текущей разработки существует 3 типа пользователей:

- Клиент - подает запрос на поиск компании, которая была бы способна оказать необходимую услугу; имеет возможность просматривать список доступных компаний;
- Компания - предоставляет некоторые услуги; осуществляет поиск клиентов среди доступной клиентской базы;
- Агентство - осуществляет регистрацию клиентов и компаний в базе;

### 2) Функциональные требования

- Хранение информации о компаниях: название, предоставляемые услуги, контактная информация, адрес главного офиса (при наличии)
- Хранение информации о клиентах: ФИО, описание необходимой услуги, контактная информация, дата запроса
- Хранение и обновление информации о статусе запроса клиента (активный/выполнен)
- Хранение и создание клиентских отзывов о качестве оказания услуги конкретной компанией
- Хранение обратной связи о подборе клиентов или компаний
- Получение информации обо всех клиентских запросах (именно: список услуг)
- Получение списка клиентов, которые желают об оказании той или иной услуги
- Получение списка компаний, оказывающих конкретную услугу
- Получение списка клиентов, которые отказались от услуги

### 3) \*Нефункциональные требования

- Запрос должен иметь дату
- Компания может оказывать разные услуги
- Клиент может требовать множество услуг
- Компания или клиент могут отказаться от оказания или получения услуг

### 4) Предварительная схема БД (сразу нормализованная)

Сущности:

**Компании:** CompanyId, Название, Контактный телефон, Адрес

**Клиенты:** ClientId, Имя, Фамилия, Телефон

**Запросы Компаний:** RequestId, CompanyId, ServiceId

**Запросы Клиентов:** RequestId, ClientId, ServiceId

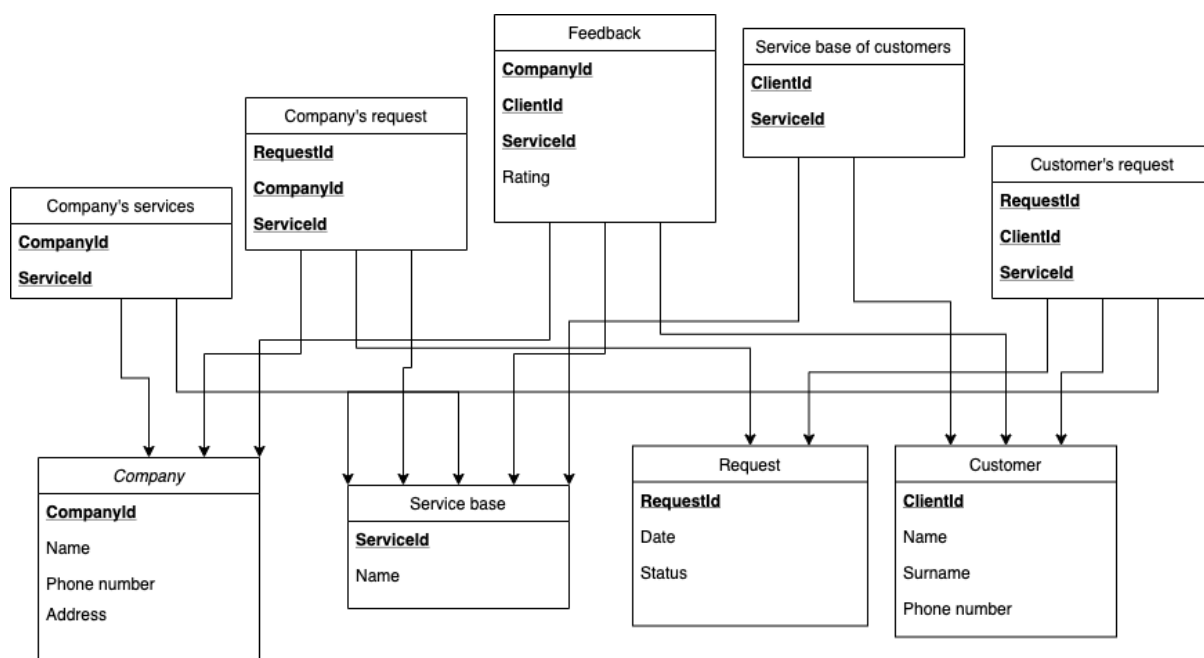
**Запросы:** RequestId, время, статус

**Feedback:** CompanyId, ClientId, ServiceId, Rating

**Услуги компаний:** CompanyId, ServiceId

**База услуг клиентов (список желаний):** ClientId, ServiceId

**База услуг:** ServiceId, Название



5) **Текстовые ограничения на данные (функциональные зависимости)**

Сфера Деятельности → Компания

Сфера Деятельности → Клиент

6) **Нормализация предварительной схемы - в 4 пункте**

7) **Ненормализованная схема и cases, в которых появляются аномалии использования**

Сущности:

**Компания:** CompanyId, название, предоставляемые услуги, контактная информация, адрес главного офиса, дата запроса, статус запроса, описание запроса

**Клиент:** ClientId, ФИО, контактная информация, дата запроса, статус запроса, описание запроса

**Отзыв:** FeedbackId, ClientId, CompanyId, оценка

**case 1 (аномалия удаления):**

При удалении запроса - мы потеряем информацию не только о самом запросе, но и о клиенте.

**case 2 (аномалия обновления):**

Поскольку один клиент может иметь несколько запросов - в таблице будет содержаться несколько записей о клиенте, которые будут дублировать его данные. Следовательно, при смене клиентом контактной информации в таблице будет содержаться строки с одним клиентом, имеющим разную контактную информацию, противоречащую друг другу.

**case 3 (аномалия добавления):**

Добавляя новую компанию или клиента неизвестно, что нужно писать в полях: дата запроса, статус запроса, описание запроса. Так же (не аномалия, просто неудобно) это ведет к неатомарности данных (в поле “предоставляемые услуги” будет набор услуг)

**8) Скрипт на SQL DDL**

```
create table Companies (  
    CompanyID number not null constraint company_pk primary  
    key,  
    Name varchar2(4000),  
    Phone number varchar2(4000),  
    Address varchar2(4000)  
);
```

```
create table Services (  
    ServiceID number not null constraint service_base_pk  
    primary key,  
    Name varchar2(4000)  
);
```

```
create table Requests (  
    RequestID number not null constraint request_pk primary  
    key,  
    Date date,  
    Status varchar2(4000)  
);
```

```
create table Customers (  
    ClientID number not null constraint customer_pk primary  
    key,  
    Name varchar2(4000),  
    Surname varchar2(4000),  
    Phone number varchar2(4000)  
);
```

```
create table CustomerRequests (  
    RequestID number not null,  
    ClientID number not null,  
    ServiceID number not null  
);
```

```
create table ClientServices (  
    ClientId number not null,  
    ServiceId number not null  
);
```

```
create table Feedback (  
    CompanyId number not null,  
    ClientId number not null,  
    ServiceId number not null,  
    Rating number  
);
```

```
create table CompaniesRequests (  
    RequestId number not null,  
    CompanyId number not null,  
    ServiceId number not null  
);
```

```
create table CompaniesServices (  
    CompanyId number not null,  
    ServiceId number not null  
);
```

## **9) Запросы, реализующие функциональные требования**

Выбрать из базы клиентов/компании, требующих набор услуг

```
SELECT
FirstName,
LastName
Phone
FROM
Customers
WHERE ID IN
(
SELECT ID
FROM CUSTOMERS JOIN ClientServices ON ID
WHERE ServiceID IN [тут сервисы]
)
```

```
SELECT
Name,
Phone
FROM
Companies
WHERE ID IN
(
SELECT ID
FROM Companies JOIN CompaniesServices ON ID
WHERE ServiceID IN [тут сервисы]
)
```

Выбрать из базы клиентов/компании, давших отказ

```
SELECT
ID,
Phone
FROM Customers JOIN Feedback ON Customers.ID = CustomerID
WHERE Rating < 3
```

```
SELECT
ID,
Name
FROM Companies JOIN Feedback ON Companies.ID = CompanyID
WHERE Rating < 3
```

Выбрать из базы все запросы клиента

```
SELECT DISTINCT
ServiceName
FROM Customers JOIN CustomerRequests ON Customers.ID = CustomerID
JOIN Services ON ServiceID = Service.ID
WHERE CustomerID = [ID нужного пользователя]
```

## 10) Группировка запросов в транзакции

Добавить новую компанию

```
BEGIN TRANSACTION;
INSERT INTO Companies
VALUES (NEWID() ,$name, $phone, $address);
COMMIT; -- Подтвердить
ROLLBACK; -- Отменить
```

Изменить набор услуг, оказываемых компанией

```
BEGIN TRANSACTION;
UPDATE CompaniesServices
SET ServiceID = $newServiceID -- заменить услугу
WHERE CompanyID = $ID AND ServiceID = $oldServiceID;
COMMIT; -- Подтвердить
ROLLBACK; -- Отменить
```

Изменить статус запроса

```
BEGIN TRANSACTION;  
UPDATE Requests  
SET Status = $newStatus -- заменить статус  
WHERE RequestID = $ID;  
COMMIT; -- Подтвердить  
ROLLBACK; -- Отменить
```