

Проектирование базы данных для маркетплейса услуг

Описание проекта

В рамках данного проекта я спроектировал базу данных для сервиса по подбору частных специалистов. Заказчики на маркетплейсе услуг сначала оставляют заявку (создают заказ на услугу, он помечается как “свободный”). В заказе они указывают тип услуги, текстовое описание, возможно прикладывают фотографии, видеозаписи, музыкальные фрагменты, любые другие файлы, возможно, геолокацию. Кроме этого они могут в заказе указать желаемый минимальный рейтинг Исполнителя. После формирования заказа, он попадает на рынок заказов к Исполнителям. Исполнители могут использовать чат с Заказчиком для обсуждения стоимости услуги, сроков её предоставления и т.д. Или могут совершить видеовызов (полезно для репетиторства или для более подробного описания проблемы). Исполнители имеют возможность отфильтровать заказы по типу услуги. Заказчик выбирает, когда заказ подтверждается специалистом, он помечается статусом “занят”. После того как услуга была предоставлена заказ помечается как завершённый. Кроме того, Заказчик и Специалист имеют возможность оставить текстовый отзыв друг о друге и поставить целочисленный рейтинг - от 1 до 5. Этот рейтинг потом учитывается при подсчете на нецелочисленного итогового рейтинга человека (от 1 до 5 либо 0, если еще нет оценок).

У клиента и у специалиста в личном кабинете доступен список созданных/выполненных им заказов. Регион, в котором выполняется заказ, подставляется автоматически, например при добавлении человеком геолокации задания. Это нужно для повышения эффективности подсчета удаленности заказчика от места выполнения услуги (чтобы считать удаленность только для заданий из, например, Санкт-Петербурга, а не для всех заданий в России).

Построение отношений

В результате предварительного проектирования были выделены следующие отношения:

- Persons — Люди.
- Services — Словарь услуг. Пример: Массаж, ремонт стиральных машин, репетиторство по математике.
- ServiceGroups — Словарь групп услуг. Пример: Здоровье и Медицина, Ремонт Бытовой Техники, Репетиторство.
- Orders — Заказы на услугу.
- Reviews — Отзывы о мастере.
- Chats — Чаты.
- Messages — Сообщения.
- Files — Вложенные файлы в заказе.
- Videocalls — Видеозвонки.

- Regions — Географический регион
- LatLons — Географические координаты (долгота и широта)
- ExpertsOrders — Исполнители заказов
- RegionsOrders — Регионы заказов
- LatLonsOrders — Точные географические координаты заказов
- Countries — Список стран.

Отношение Persons

Атрибуты:

- PersonId
- Name
- Rating
- PersonType

Функциональные зависимости:

- PersonId -> Name, Rating, PersonType

Ключи:

- PersonId

Нормализация:

1. Отношение находится в 1НФ, т.к. в отношении нет повторяющихся групп, все атрибуты атомарны и у отношения есть ключ.
2. Отношение находится в 2НФ, потому что отношение находится в 1НФ и неключевые атрибуты функционально зависят от ключа .
3. Отношение находится в 3НФ, т.к. оно находится в 2НФ и неключевые атрибуты не транзитивно зависят от ключей.
4. По первой теореме Дейта-Фейгина отношение находится в 5НФ так как оно находится в 3НФ и PersonId единственный и простой ключ.

Отношение ServiceGroup

Атрибуты:

- ServiceGroupId
- Name

Функциональные зависимости:

- ServiceGroupId -> Name

Ключи:

- ServiceGroupId

Отношение Reviews

Атрибуты:

- ReviewId
- Text

- Rating (целое число от 1 до 5 включительно)
 - AuthorId (автор отзыва)
 - RecipientId (получатель отзыва)
 - CreationTime
- Функциональные зависимости:
- ReviewId -> Text, Rating, AuthorId, RecipientId, CreationTime
- Ключи:
- ReviewId

Отношение Videocalls

Атрибуты:

- VideallId
- StartDate (начало звонка)
- EndDate (конец звонка)
- PersonId1
- PersonId2

Функциональные зависимости:

- VideallId -> StartDate, EndDate, PersonId1, PersonId2

Ключи:

- VideallId

Отношение Chats

Атрибуты:

- ChatId
- PersonId1
- PersonId2
- ChatStatus

Функциональные зависимости:

- ChatId -> PersonId1, PersonId2, ChatStatus

Ключи:

- ChatId

Отношение Messages

Атрибуты:

- MessageId
- Text
- ChatId
- SenderId
- SendingTime

Функциональные зависимости:

- MessageId -> Text, ChatId, SenderId, SendingTime

Ключи:

- MessageId

Отношение Regions

Атрибуты:

- RegionId
- Name
- LatLonId (географический центр региона, примерный)
- CountryId

Функциональные зависимости:

- RegionId -> Name, LatLonId, CountryId

Ключи:

- RegionId

Отношение LatLons

Атрибуты:

- LatLonId
- Lat (широта)
- Lon (долгота)

Функциональные зависимости:

- LatLonId -> Lat, Lon

Ключи:

- LatLonId

Отношение Files

Атрибуты:

- FileId
- Location (path to file on server)
- OrderId (заказ к которому файл приложен)

Функциональные зависимости:

- FileId -> Location, OrderId

Ключи:

- FileId

Нормализация:

- Отношение находится в 1НФ, т.к. в отношении нет повторяющихся групп, все атрибуты атомарны и у отношения есть ключ.
- Отношение находится в 2НФ, потому что отношение находится в 1НФ и неключевые атрибуты функционально зависят от ключа .
- Отношение находится в 3НФ, т.к. оно находится в 2НФ и неключевые атрибуты не транзитивно зависят от ключей.

- По первой теореме Дейта-Фейгина отношение находится в 5НФ так как оно находится в 3НФ и Fileld единственный и простой ключ.

Отношение Services

Атрибуты:

- Serviceld
- Name
- ServiceGroupld

Функциональные зависимости:

- Serviceld -> Name, ServiceGroupld

Ключи:

- Serviceld

Отношение Orders

Атрибуты:

- Orderld
- Text
- ExpertRating (минимальный уровень исполнителя)
- CreationTime (дата и время создания заказа)
- AcceptanceTime (дата и время согласия исполнителя выполнить заказ)
- CompletionTime (дата и время выполнения заказа)
- OrderStatus
- Clientld (заказчик)
- Serviceld (тип услуги)

Функциональные зависимости:

- Orderld -> ExpertRating, CreationTime, AcceptanceTime, CompletionTime, OrderStatusld, Clientld, Serviceld

Ключи:

- Orderld

Нормализация:

1. Отношение находится в 1НФ, т.к. в отношении нет повторяющихся групп, все атрибуты атомарны и у отношения есть ключ.
2. Отношение находится в 2НФ, потому что отношение находится в 1НФ и неключевые атрибуты функционально зависят от ключа .
3. Отношение находится в 3НФ, т.к. оно находится в 2НФ и неключевые атрибуты не транзитивно зависят от ключей.
4. По первой теореме Дейта-Фейгина отношение находится в 5НФ так как оно находится в 3НФ и Serviceld единственный и простой ключ.

Отношение ExpertsOrders

Атрибуты:

- OrderId
 - PersonId (Исполнитель)
- Функциональные зависимости:
- OrderId -> PersonId
- Ключи:
- OrderId

Отношение RegionsOrders

- Атрибуты:
- OrderId
 - RegionId
- Функциональные зависимости:
- OrderId -> RegionId
- Ключи:
- OrderId

Отношение LatLonsOrders

- Атрибуты:
- OrderId
 - LatLonId
- Функциональные зависимости:
- OrderId -> LatLonId
- Ключи:
- OrderId

Отношение Countries

- Атрибуты:
- CountryId
 - Name
- Функциональные зависимости:
- CountryId -> Name
- Ключи:
- CountryId

Модель сущность-связь

Связи по умолчанию: Optional
Атрибуты по умолчанию: Mandatory

<i>id</i>	serial
<i>time</i>	timestampzt
<i>review_raiting</i>	int
<i>person_raiting</i>	float
<i>lat</i>	float
<i>lon</i>	float
<i>person_type</i>	Custom Enum <i>person_type</i>
<i>review_text</i>	varchar(200)
<i>person_name</i>	varchar(80)
<i>messages_text</i>	varchar(200)
<i>region_name</i>	varchar(100)
<i>chat_status</i>	Custom Enum <i>chat_status</i>
<i>order_status</i>	Custom Enum <i>order_status</i>
<i>country_name</i>	varchar(60)
<i>file_location</i>	varchar(90)
<i>service_name</i>	varchar(50)
<i>services_group_name</i>	varchar(50)

Определения таблиц

Для реализации проекта использовалась СУБД PostgreSQL 14.1. Определения таблиц и их индексов приведено в файле `ddl.sql`.

Тестовые данные

Скрипт для добавления тестовых данных приведен в файле `data.sql`.

Запросы на получение данных

В рамках проекта были реализованы следующие запросы:

- 1. `getAllTaken` — Получение по `id` пользователя всех заказов, которые он взялся исполнять
- 2. `getAllCreated` — Получение по `id` пользователя, всех созданных им заказов
- 3. `getAllAvailable` — Получение по рейтингу пользователя список всех доступных ему для исполнения заказов
- 4. `getAllInRegion` — Получение по `id` региона всех свободных в нем заказов
- 5. `getAllTakenTypes` — Получение по `id` пользователя, названия всех типов услуг, за которые он брался
- 6. `getReviewStat` — Получение по `id` пользователя, статистики по всем оценкам о нем
- 7. `getOrderTypesStat` — Получение статистики о количестве заказов, по типу услуги
- 8. `getRegionStat` — Получение статистики о количестве заказов, по региону
- 9. `getFilesInOrder` — Получение файлов по номеру заказа
- 10. `getAllPersonChats` — Получение по `id` пользователя, всех его переписок
- 11. `getAllChatMessages` — Получение по `id` чата, всех его сообщений
- 12. `getPersonVideoCallsHistory` — Получение по `id` пользователя, всех его видеозвонков
- 13. `getAllPersonReviews` — Получение по `id` пользователя, всех отзывов о нем
- 14. `getAllPersonReviewsRatings` — Получение по `id` пользователя, всех оценок в отзывах о нем

Запросы на получение данных приведены в файле `selects.sql`.

Запросы на изменение данных

В рамках проекта были реализованы следующие запросы:

- 1.1 `createPerson` — создаем человека
- 1.2 `createReview` — создаем отзыв без текста
- 1.3 `createFullReview` — создаем отзыв с текстом
- 1.4 `createVideoCall` — создаем запись о видеозвонке
- 2.1 `deletePerson` — удаление пользователя по заданному `id` при помощи триггеров реализовано каскадированное удаление
- 2.2 `deleteOrder` — удаление заказа по заданному `id` при помощи триггеров реализовано каскадированное удаление
- 2.3 `deleteChat` — удаление чата по заданному `id` при помощи триггеров реализовано каскадированное удаление

- 2.4 deleteMessage — удаление сообщения по заданному id при помощи триггеров реализовано каскадированное удаление
- 3.1 CreateOrder — создать заказ на услугу
- 3.2 TakeOrder — принять заказ в работу каким-то человеком
- 3.3 GiveUpOrder — отказаться от выполнения заказа
- 3.4 CompleteOrder — пометить заказ как завершённый
- 3.5 AddLatLonToOrder — добавить геолокацию в заказ
- 3.6 AddRegionToOrder — добавить регион в заказ
- 3.7 AddExpertRatingToOrder — добавить в заказ нижнюю границу рейтинга для исполнителя
- 3.8 CreateLatLon — создать геолокацию
- 3.9 CreateChat — создать чат между двумя людьми
- 3.10 CloseChat — закрыть чат (забанить человека)
- 3.11 SendMessage — отправить сообщение

Запросы на изменение данных, хранимые процедуры и триггеры приведены в файле `updates.sql`.