

Экзаменационный проект
по дисциплине
Проектирование баз данных
студента гр. **М34361**
Щукина Георгия Алексеевича
по теме
**Сервис найма команд для
многопользовательских онлайн
видеоигр**

Описание проекта

В рамках проекта реализуется база данных для сервиса, предоставляющего услуги по обучению/продвижению в командных онлайн играх, в группе проверенных опытных игроков с выбранными покупателем параметрами, а также прием команд-исполнителей в наш список, выдача им заказов, обеспечение коммуникации с заказчиком и качества исполнения.

Сам сервис предоставляет платформу для формирования команд и координации игровых сессий в различных многопользовательских онлайн играх.

Построение отношений

В результате предварительного проектирования были выделены следующие отношения:

- <<Users>> - описывает пользователя сервиса.
- <<Languages>> - описывает поддерживаемые сервисом языки.
- <<Games>> - описывает для каких игр и на каких платформах предоставляются услуги.
- <<Teams>> - описывает команду-исполнителя, лидер/представитель команды считается пользователем.
- <<Requests>> - описывает доступные для назначения командам заказы, включает описание заказа.
- <<Orders>> - описывает выполненный заказ или заказ в исполнении.
- <<Feedback>> - описывает оставленные пользователями отзывы о командах.
- <<SupportTickets>> - описывает обращения в поддержку с вопросами/претензиями/поправками по конкретному заказу.

Отношение <<Users>>

Атрибуты:

- <<UserId>> - идентификатор пользователя.
- <<UserName>> - имя пользователя.
- <<UserLanguageId>> - язык пользователя.
- <<UserContactInfo>> - контактная информация.

Функциональные зависимости

- UserId -> UserName
- UserId -> UserLanguageId
- UserId -> UserContactInfo

Ключи:

- UserId

Отношение <<Languages>>

Атрибуты:

- <<LanguageId>> - идентификатор языка.
- <<LanguageName>> - название языка.
- <<LanguageCode>> - код страны языка.

Функциональные зависимости

- LanguageId-> LanguageName
- LanguageId-> LanguageCode

Ключи:

- LanguageId

Отношение <<Games>>

Атрибуты:

- <<GameId>> - идентификатор игры.
- <<GameName>> - название игры.
- <<PlatformName>> - название платформы.

Функциональные зависимости

- GameId-> GameName
- GameId-> PlatformName

Ключи:

- GameId

Отношение <<Teams>>

Атрибуты:

- <<TeamId>> - идентификатор команды.
- <<TeamName>> - имя команды.
- <<TeamLeaderUserId>> - идентификатор пользователя-представителя команды.
- <<TeamHoursStart>> - время начала доступности команды.
- <<TeamHoursEnd>> - время конца доступности команды.
- <<TeamSize>> - количество человек в команде.

Функциональные зависимости

- TeamId -> TeamName
- TeamId -> TeamLeaderUserId
- TeamId -> TeamHoursStart
- TeamId -> TeamHoursEnd
- TeamId -> TeamSize

Ключи:

- TeamId

Отношение <<Requests>>

Атрибуты:

- <<RequestId>> - идентификатор заявки.

- <<UserId>> - идентификатор пользователя, сделавшего заявку.
- <<RequestedTime>> - желаемое время исполнения.
- <<RequestedGameId>> - желаемая игра и платформа
- <<RequestedTeamSize>> - желаемое количество игроков в команде.

Функциональные зависимости

- RequestId-> UserId
- RequestId-> RequestedTime
- RequestId-> RequestedGameId
- RequestId-> RequestedTeamSize

Ключи:

- RequestId

Отношение <<Orders>>

Атрибуты:

- <<OrderId>> - идентификатор принятого заказа.
- <<RequestId>> - идентификатор выполняемой заявки.
- <<OrderStatus>> - статус заказа.
- <<AssignedTeamId>> - идентификатор назначенной команды.

Функциональные зависимости

- OrderId -> RequestId
- OrderId -> OrderStatus
- OrderId -> AssignedTeamId

Ключи:

- OrderId

Нормализация:

1НФ: В отношении нет повторяющихся групп, все атрибуты атомарны, присутствует ключ.

2НФ: Неключевые атрибуты, согласно ФЗ, зависят от ключа в целом.

3НФ: Неключевые атрибуты, согласно ФЗ, напрямую зависят от ключа.

НФБК: В каждой нетривиальной ФЗ левая часть - надключ.

4НФ: По второй теореме Дейта-Фейгина, так как отношение находится в НФБК и существует простой ключ OrderId, то отношение в 4НФ.

5НФ: По первой теореме Дейта-Фейгина, так как отношение находится в 3НФ и все ключи (OrderId) простые, то отношение в 5НФ.

Отношение <<Feedback>>

Атрибуты:

- <<UserId>> - идентификатор пользователя.
- <<TeamId>> - идентификатор команды для которой пишется отзыв.
- <<Rating>> - выставленная оценка от 1 до 5.

Функциональные зависимости

- UserId, TeamId -> Rating

Ключи:

- (UserId, TeamId)

Нормализация:

1НФ: В отношении нет повторяющихся групп, все атрибуты атомарны, присутствует ключ.

2НФ: Неключевой атрибут Rating, согласно ФЗ, зависит от ключа в целом.

3НФ: Неключевой атрибут Rating, согласно ФЗ, напрямую зависит от ключа.

НФБК: В каждой нетривиальной ФЗ левая часть - надключ.

4НФ: Покажем, что не существует нетривиальной МЗ вида $X \rightarrow Y \mid Z$. Для равных атрибутов X всегда можно привести разные атрибуты из Y и Z . Также можем перебрать все варианты таких МЗ и увидеть, что они тривиальные. Следовательно не можем декомпозировать на два отношения и имеем 4НФ.

5НФ: Перебрав все декомпозиции отношения видим, что все они не могут быть 3С. $\pi X_1(R) \bowtie \pi X_2(R) \bowtie \dots \bowtie \pi X_n(R)$ может не совпадать с нашим отношением R для каждого разбиения, отношение в 5НФ.

Отношение <<SupportTickets>>

Атрибуты:

- <<TicketId>> - идентификатор обращения.
- <<TicketStatus>> - статус обращения.
- <<OrderId>> - идентификатор заказа по которому требуется поддержка.
- <<Comment>> - комментарий пользователя.

Функциональные зависимости

- TicketId \rightarrow TicketStatus
- TicketId \rightarrow OrderId
- TicketId \rightarrow Comment

Ключи:

- TicketId

Нормализация:

1НФ: В отношении нет повторяющихся групп, все атрибуты атомарны, присутствует ключ.

2НФ: Неключевые атрибуты, согласно ФЗ, зависят от ключа в целом.

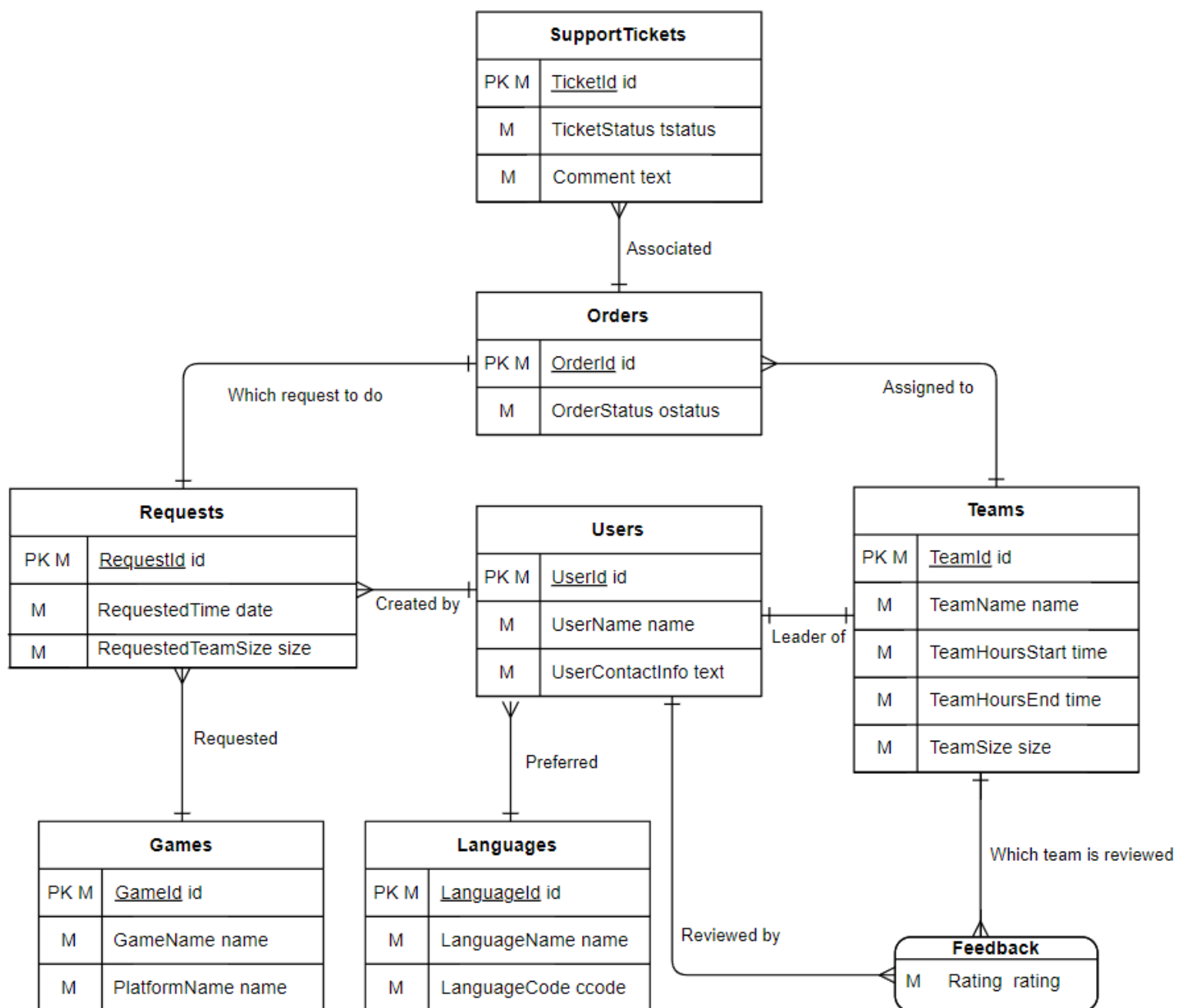
3НФ: Неключевые атрибуты, согласно ФЗ, напрямую зависят от ключа.

НФБК: В каждой нетривиальной ФЗ левая часть - надключ.

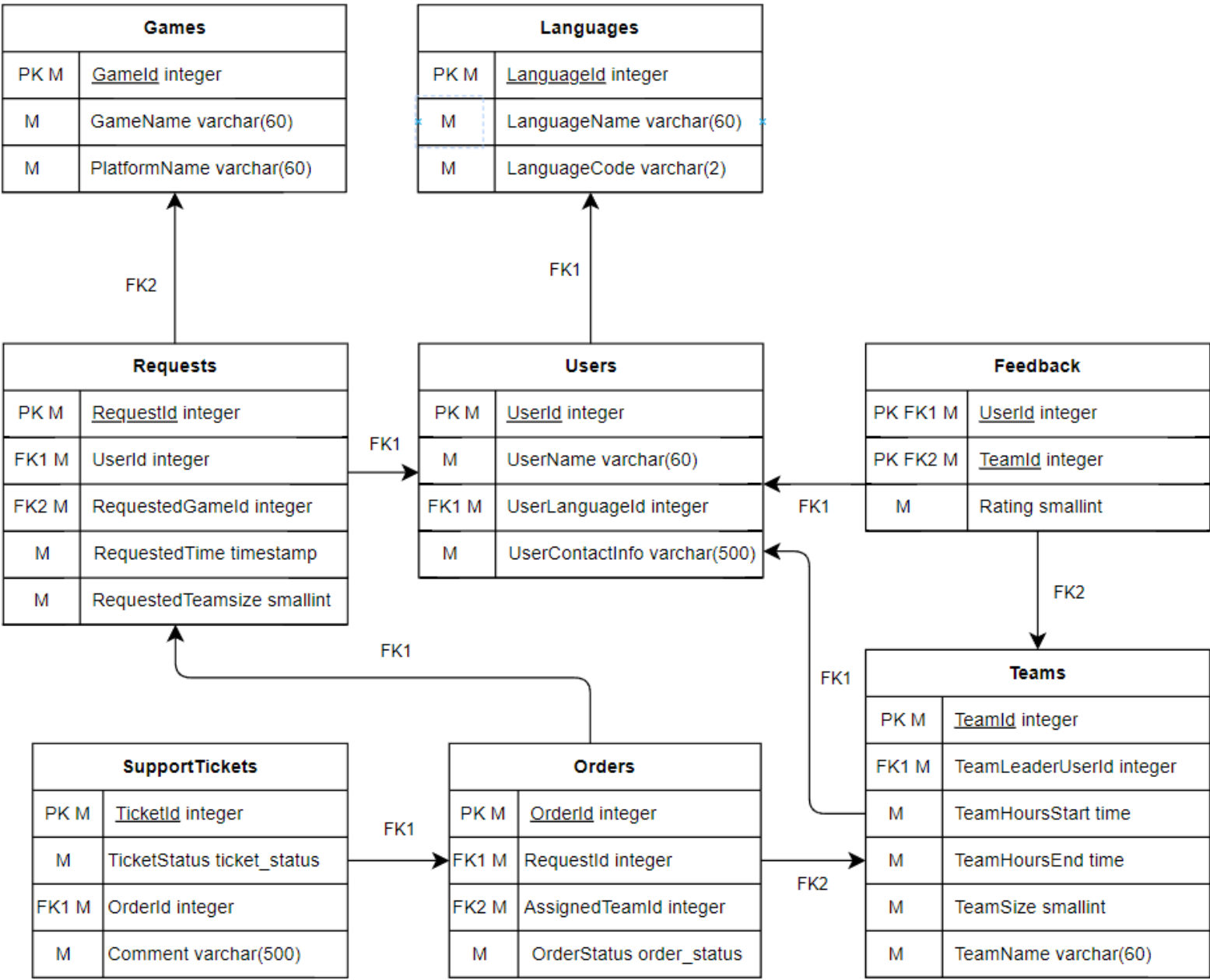
4НФ: По второй теореме Дейта-Фейгина, так как отношение находится в НФБК и существует простой ключ TicketId, то отношение в 4НФ.

5НФ: По первой теореме Дейта-Фейгина, так как отношение находится в 3НФ и все ключи (TicketId) простые, то отношение в 5НФ.

Модель сущность-связь



Физическая модель



При построении физической модели использовалось следующее отображение доменов в типы:

Домен	Тип
id	integer
name	varchar (60)
text	varchar (500)
ccode	varchar (2)

time	time
size	smallint
rating	smallint
date	timestamp
ostatus	enum('In Progress', 'Cancelled', 'Completed') (order_status)
tstatus	enum('Open', 'Completed', 'Refuted') (ticket_status)

Определения схемы

Для реализации проекта использовалась СУБД PostgreSQL 15.4. Определения таблиц и их индексов приведено в файле `ddl.sql` (Приложение 1).

Тестовые данные

Скрипт для добавления тестовых данных приведен в файле `insert.sql` (Приложение 2).

Запросы на получение данных

В рамках проекта были реализованы следующие запросы:

- Статистика команд - название команды, средний рейтинг и кол-во выполненных заказов. Отсортирована по убыванию числа заказов.
- Свободные сегодня заявки - заявки на текущую дату, для которых нет закрепленной команды.
- Открытые обращения для заказов в процессе - обращения статуса "Открыто" в поддержку для заказов в процессе исполнения.
- Заявки которые не могут быть выполнены ни одной командой - заявки начиная с текущей даты для которых не нашлось команды подходящего размера и с подходящим временем работы.

Для реализации запросов были созданы вспомогательные представления:

- <<TeamRatings>> - представление считающее средний рейтинг команды по отзывам.
- <<RequestsAvailableToday>> - заявки доступные сегодня

Запросы на получение данных и вспомогательные представления приведены в файле `select.sql` (Приложение 3).

Запросы на изменение данных

В рамках проекта были реализованы следующие запросы:

- <<assign_request_to_any_team>> - назначение заказа любой подходящей команде.
- <<complete_order>> - отметить выполнение заказа.
- <<open_support_ticket>> - создание нового обращения в поддержку.
- <<refute_support_ticket>> - отмена существующего обращения в поддержку.
- <<close_support_ticket>> - завершение существующего обращения в поддержку.

Запросы на изменение данных, хранимые процедуры и триггеры приведены в файле `update.sql` (Приложение 4).