Лабораторная работа №1. Проектирование архитектуры БД.

Вариант №2

Выполнила студентка группы М3212

Авсюкевич Анастасия

Задачи:

1) Провести анализ функционала сайта или портала в выбранной предметной области с позиции работы с данными, выделить сущности, их атрибуты и связи между сущностями.

2) Спроектировать архитектуру БД для выбранной темы в виде ER-diagram в нотации IDEF1X или UML.

Порядок выполнения работы:

1) Выбрать один из предложенных ниже вариантов предметных областей или

согласовать с преподавателем другой вариант.

2) Провести анализ функционала сайта или портала в выбранной предметной области с позиции работы с данными, выделить сущности, их атрибуты и связи между сущностями.

3) Спроектировать архитектуру БД для выбранной темы в виде ER-diagram в нотации IDEF1X или UML.

4) Предоставить отчёт, включить в него следующие данные:

– краткое описание предметной области и портала, в том числе пояснение,

охватывает ли моделирование все данные или только те, которые относятся к

некоторым бизнес-процессам (указать к каким);

– модель данных, построенная с учетом заданных требований;

– обоснование нахождения модели данных в 3НФ (или выше по желанию).

Требования к архитектуре БД:

1) Минимум 3 НФ.

2) Минимум 6 таблиц, максимум 12 таблиц.

3) Должны быть указаны ключи.

4) Должны быть разные связи, в том числе многие ко многим и один ко многим.

5) Типы атрибутов должны быть указаны.

6) Должна существовать таблица пользователей.

Задание моего варианта (№2):

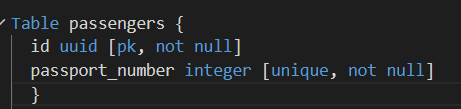
2) Выбор и покупка билетов на поезд https://www.rzd.ru/

Для выполнения проектирования предметной областью, которую я изучала и моделировала, был выбран сайт РЖД, а если конкретней - функция оформления заказа (выбора и покупки билета). Так как сайт имеет довольно обширный функционал: начиная от возможности просмотра различной информации о данной транспортной организации, заканчивая покупкой услуги «Душ в поезде» в разных тарифах, я решила сузить предметную область и сосредоточиться на конкретном бизнес-процессе – покупке билета через сайт (то есть онлайн). В соответствие с выбранным мной объектом проектирования, были выделены данные, которые я посчитала нужными для работы данного функционала.

В итоге для задачи «выборка и покупка билетов на поезд на сайте РЖД» мне удалось выделить следующие сущности, их атрибуты и связи:

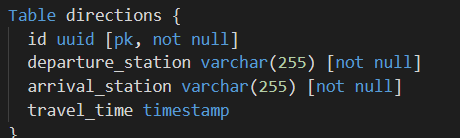
Сущности и их атрибуты:

1. Passengers



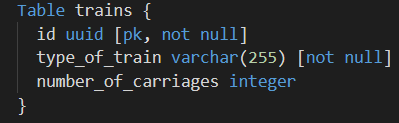
Сущность passengers (пассажиры), в которой хранится основная информация, необходимая для покупки билета на поезд (то есть для оформления заказа). Но так как заказывать билет может один человек для другого (того, кто будет пассажиром), я решила вынести пассажира в отдельную сущность.

1. Directions



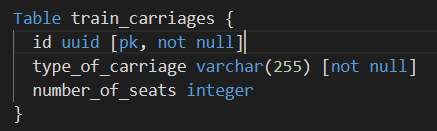
Сущность directions (пути следования). Данная сущность содержит следующие атрибуты, отражающие ее суть: айди, станция отправления, станция прибытия, время в пути.

1. Trains



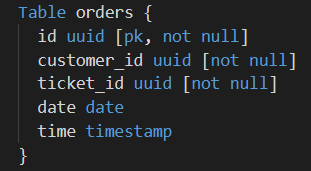
Сущность trains (поезда) так же является одной из ключевых. Я решила, что ей должны быть присущи следующие атрибуты: айди, тип поезда (например, «Сапсан»), кол-во вагонов. Соответствующие типы данных атрибутов представлены на скриншоте.

1. Train\_carriages



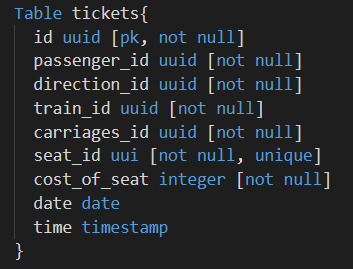
Сущность train\_carriages (вагоны), зависящая от поезда (состава, к которому оно прикрепляется), имеет следующие атрибуты: айди, тип вагона (например, «Купе»), кол-во мест.

1. Orders



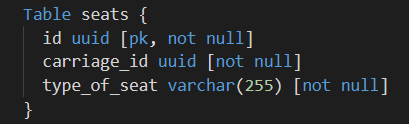
«Самая важная», если в контексте проектирования БД корректно так выразиться, сущность – orders (заказы), с помощью которой мы может узнать основную информацию о выбранном мной бизнес-процессе. Данная сущность содержит следующие атрибуты: айди, айди покупателя (FK), айди билета (FK), дату и время оформления.

1. Tickets



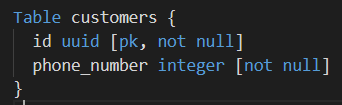
Сущность tickets (билеты), которая связывает все остальные сущности, имеет следующие атрибуты: айди, айди пассажира (FK), айди пути следования, айди поезда, айди вагона, айди места, стоимость места, зависящая от конкретных атрибутов билета, дату и время приобретения.

1. Seats



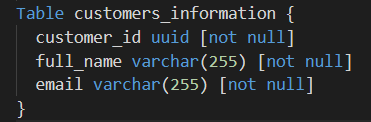
Сущность seats (места), зависящая от вагона, которому она принадлежит, имеет следующие атрибуты: айди, айди вагона (FK), тип места (например: «Плацкартное», «Сидячее» и тд)

1. Customers



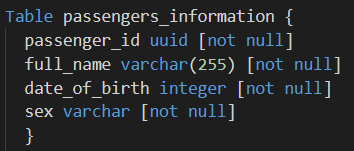
Сущность customers (покупатели) – требуемая по ТЗ «таблица пользователей», была выделена отдельно, так как, как уже было сказано выше, заказывать билет может один человек для другого (того, кто будет пассажиром). Данная сущность имеет следующие атрибуты: айди и номер телефона.

1. Customers\_information



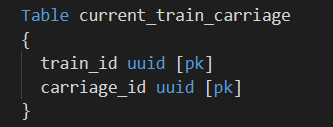
Вспомогательная таблица, необходимая для избавления от транзитивной зависимости. Данная сущность является «продолжением» сущности Customers, в ней хранится ФИО и эл. почта покупателя – информация, нужная при оформлении заказа.

1. Passengers\_info



Вспомогательная таблица, необходимая для избавления от транзитивной зависимости. Данная сущность является «продолжением» сущности Passengers, в ней хранится информация, нужная при оформлении заказа.

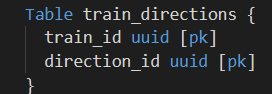
1. Current\_train\_carriage



Вспомогательная таблица, необходимая для связи многие-ко-многим между сущностями «Поезд» и «Вагон».

В качестве атрибутов содержит айдишники поезда и вагона.

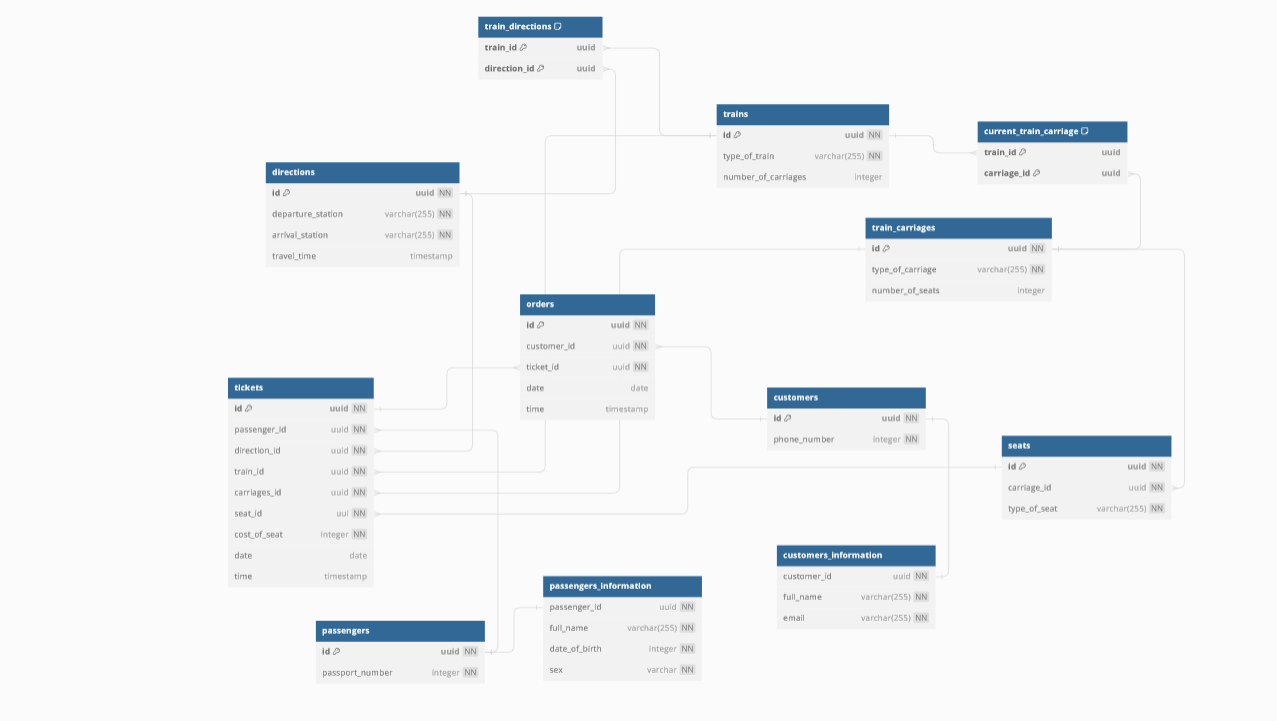
1. Train\_directions



Вспомогательная таблица, необходимая для связи многие-ко-многим между сущностями «Поезд» и «Путь следования».

В качестве атрибутов содержит айдишники поезда и направления пути.

На основе этих сущностей и их связей мне удалось спроектировать ER-диаграмму. Вот что получилось:



Обоснование нахождения модели данных в 3НФ:

Начнем по порядку…

1. БД находится в 1 НФ, так как по определению:

Отношение находится в **первой нормальной форме**, если все его атрибуты являются простыми.

На схеме очень нагляден факт того, что все атрибуты действительно атомарны и нет повторения строк в таблице.

! У меня атрибут full\_name является единой атомарной единицей

1. БД находится во 2 НФ, так как по определению:

Отношение находится во **второй нормальной форме**, если:

* Оно находится в первой нормальной форме
* Все неключевые атрибуты таблицы зависят от первичного ключа

Первое требование уже выполнено, так как в таблице:

* Каждая ячейка хранит только одно значение
* Все данные в одной колонке одного типа
* Каждая запись отличается от других записей

Второе требование так же автоматически выполнено, так как таблицы не имеют составных первичных ключей (все они простые), и все неключевые атрибуты соответственно полно зависят от них.

1. БД находится в третьей нормальной форме, так как по определению:

Отношение находится **в третьей нормальной форме**, если оно находится во второй нормальной форме, и ни один неключевой атрибут не находится в транзитивной функциональной зависимости от потенциального ключа.

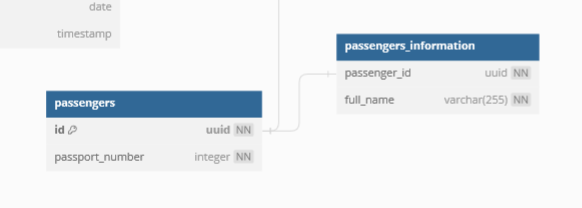
! каждый столбец, не являющийся ключом, должен зависеть только от первичного ключа.

Первое требование уже выполнено, так как:

* Таблица соблюдает требования первой нормальной формы
* Все неключевые атрибуты таблицы зависят от первичного ключа

Выполним второе требование:

Я разделила таблицы на две для предотвращения транзитивной зависимости:



И аналогично с сущностью Customers:

