

ГУАП

КАФЕДРА № 41

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

Старший преподаватель

должность, уч. степень, звание

подпись, дата

Б.К. Акопян

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3

ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

по курсу: БАЗЫ ДАННЫХ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. № 4217

подпись, дата

Д.М. Никитин

инициалы, фамилия

Санкт-Петербург 2024

1. Цель работы: разработка инфологической и даталогической моделей реляционной БД.

2. Вариант 24:

Информационная система железнодорожной пассажирской станции

Работников железнодорожной станции можно подразделить на водителей подвижного состава, диспетчеров, ремонтников подвижного состава, путей, кассиров, работников службы подготовки составов, справочной службы и других, которые административно относятся каждый к своему отделу. Каждая из перечисленных категорий работников имеет уникальные атрибуты-характеристики, определяемые профессиональной направленностью. В отделах существует разбиение работников на бригады. Отделы возглавляются начальниками, которые представляют собой администрацию железнодорожной станции. В функции администрации входит планирование маршрутов, составление расписаний, формирование кадрового состава железнодорожной станции. За каждым локомотивом закрепляется локомотивная бригада. За несколькими локомотивами закрепляется бригада техников-ремонтников, выполняющая рейсовый и плановый техосмотр (по определенному графику), ремонт, техническое обслуживание. Водители локомотивов обязаны проходить каждый год медосмотр, не прошедших медосмотр необходимо перевести на другую работу. Локомотив должен своевременно осматриваться техниками и ремонтниками и при необходимости ремонтироваться. Подготовка к рейсу включает в себя техническую часть (рейсовый техосмотр, мелкий ремонт) и обслуживающую часть (уборка вагонов, запас продуктов питания и т.п.). В расписании указывается тип поезда (скорый, пассажирский . . .), номер поезда, дни и время отправления и прибытия, маршрут (начальный и конечный пункты назначения, основные узловые станции), стоимость билета. Билеты на поезд можно приобрести заранее или забронировать в железнодорожных кассах. До отправления поезда, если есть необходимость, билет можно вернуть. Отправление поездов может быть задержано из-за опозданий поездов,

погодных условий, технических неполадок. Железнодорожные маршруты можно разделить на следующие категории: внутренние, международные, туристические, специальные маршруты. Пассажиры могут сдавать свои вещи в багажное отделение.

3. Инфологическая модель:

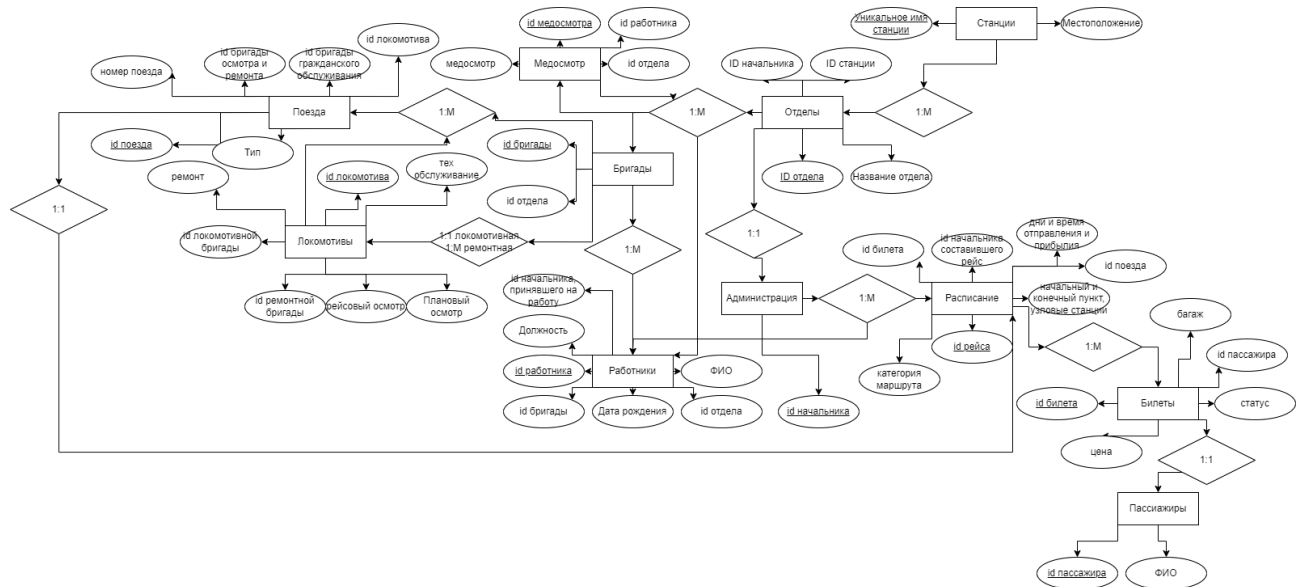


Рисунок 1 – Инфологическая модель

4. Описание сущностей, атрибутов и связей:

1. Stations (Станции)

Описание: Эта таблица хранит информацию о железнодорожных станциях.

Атрибуты:

station_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор станции.

location (VARCHAR(45)): Местоположение станции.

Связи:

Связана с таблицей Departments, где каждая станция может иметь несколько департаментов.

2. Teams (Команды)

Описание: Эта таблица хранит информацию о командах, которые могут заниматься различными задачами, такими как ремонт или обслуживание.

Атрибуты:

team_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор команды.

department_id (INT, FOREIGN KEY): Внешний ключ, который ссылается на department_id из таблицы Departments, указывая на департамент, к которому относится команда.

Связи:

Связана с таблицей Departments (департамент) через внешний ключ department_id.

Связана с таблицами Locomotives, Trains как команда, занимающаяся ремонтом или обслуживанием.

3. Employees (Сотрудники)

Описание: Содержит информацию о сотрудниках, которые работают на станции.

Атрибуты:

employee_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор сотрудника.

full_name (VARCHAR(130)): Полное имя сотрудника.

post (VARCHAR(130)): Должность сотрудника.

date_of_birth (DATE): Дата рождения сотрудника.

team_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Teams, указывающий, к какой команде принадлежит сотрудник.

Связи:

Связана с таблицей Teams через внешний ключ team_id.

Сотрудники могут также быть администраторами в таблице Administration.

Связана с таблицей Medical_examinations для хранения медицинских осмотров сотрудников.

4. Administration (Администрация)

Описание: Содержит информацию об администраторах станций.

Атрибуты:

administrator_id (INT, AUTO_INCREMENT, PRIMARY KEY):

Уникальный идентификатор администратора.

employee_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Employees, указывающий на сотрудника, который является администратором.

Связи:

Связана с таблицей Employees, откуда администратор выбирается.

Администратор может управлять департаментом (Departments) и участвовать в расписании (Schedule).

5. Departments (Департаменты)

Описание: Департамент управляет различными операциями на станциях.

Атрибуты:

department_id (INT, AUTO_INCREMENT, PRIMARY KEY):

Уникальный идентификатор департамента.

department_name (VARCHAR(45)): Название департамента.

station_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Stations, указывающий, к какой станции относится департамент.

administrator_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Administration, указывающий на администратора, ответственного за департамент.

Связи:

Связана с таблицей Stations через внешний ключ station_id.

Связана с таблицей Administration через внешний ключ administrator_id.

6. Medical_examinations (Медицинские осмотры)

Описание: Содержит информацию о медицинских осмотрах сотрудников.

Атрибуты:

medical_examination_id (INT, AUTO_INCREMENT, PRIMARY KEY):

Уникальный идентификатор медицинского осмотра.

employee_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Employees, указывающий на сотрудника.

medical_examination (VARCHAR(500)): Описание медицинского осмотра.

Связи:

Связана с таблицей Employees, указывая, к какому сотруднику относится данный медицинский осмотр.

7. Locomotives (Локомотивы)

Описание: Хранит информацию о локомотивах, их ремонтах и осмотрах.

Атрибуты:

locomotive_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор локомотива.

repair_team_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Teams, указывающий на команду, отвечающую за ремонт.

repair (VARCHAR(500)): Информация о ремонте.

regular_inspection (DATETIME): Дата последнего регулярного осмотра.

scheduled_inspection (DATETIME): Дата запланированного осмотра.

maintenance (DATETIME): Дата последнего обслуживания.

locomotive_crew_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Teams, указывающий на команду машинистов.

Связи:

Связана с таблицей Teams через внешний ключ для команды, занимающейся ремонтом.

Также связана с другой командой машинистов через locomotive_crew_id.

8. Trains (Поезда)

Описание: Содержит информацию о поездах, их типе и назначении.

Атрибуты:

train_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор поезда.

type (VARCHAR(45)): Тип поезда (например, пассажирский, грузовой).

train_number (VARCHAR(45)): Уникальный номер поезда.

repair_team_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Teams, указывающий на команду, занимающуюся ремонтом.

civil_service_team_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Teams, указывающий на команду, обслуживающую поезд.

locomotive_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Locomotives, указывающий на локомотив.

Связи:

Связана с таблицей Teams через два внешних ключа (ремонтная команда и обслуживающая команда).

Также связана с таблицей Locomotives, указывая, какой локомотив закреплен за поездом.

9. Schedule (Расписание)

Описание: Содержит расписание поездов.

Атрибуты:

flight_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор рейса.

category (VARCHAR(45)): Категория рейса (например, регулярный, чартерный).

administrator_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Administration, указывающий на администратора, ответственного за расписание.

departure (DATETIME): Время отправления.

arrival (DATETIME): Время прибытия.

train_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Trains, указывающий на поезд.

route (VARCHAR(500)): Описание маршрута.

Связи:

Связана с таблицей Trains для определения поезда.

Связана с таблицей Administration, указывая на администратора, который отвечает за рейс.

10. Passengers (Пассажиры)

Описание: Хранит информацию о пассажирах.

Атрибуты:

passenger_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор пассажира.

full_name (VARCHAR(45)): Полное имя пассажира.

Связи:

Связана с таблицей Tickets, которая хранит информацию о билетах пассажиров.

11. Tickets (Билеты)

Описание: Содержит информацию о билетах пассажиров.

Атрибуты:

ticket_id (INT, AUTO_INCREMENT, PRIMARY KEY): Уникальный идентификатор билета.

luggage (TINYINT(1)): Указывает, есть ли у пассажира багаж (1 - да, 0 - нет).

passenger_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Passengers, указывающий на пассажира, которому принадлежит билет.

status (VARCHAR(45)): Статус билета (например, "активный", "использованный").

price (INT): Цена билета.

flight_id (INT, FOREIGN KEY): Внешний ключ, ссылающийся на таблицу Schedule, указывающий на рейс, на который был куплен билет.

Связи:

Связана с таблицей Passengers через внешний ключ passenger_id.

Связана с таблицей Schedule через внешний ключ flight_id, указывая, на какой рейс оформлен билет.

5. Даталогическая модель:

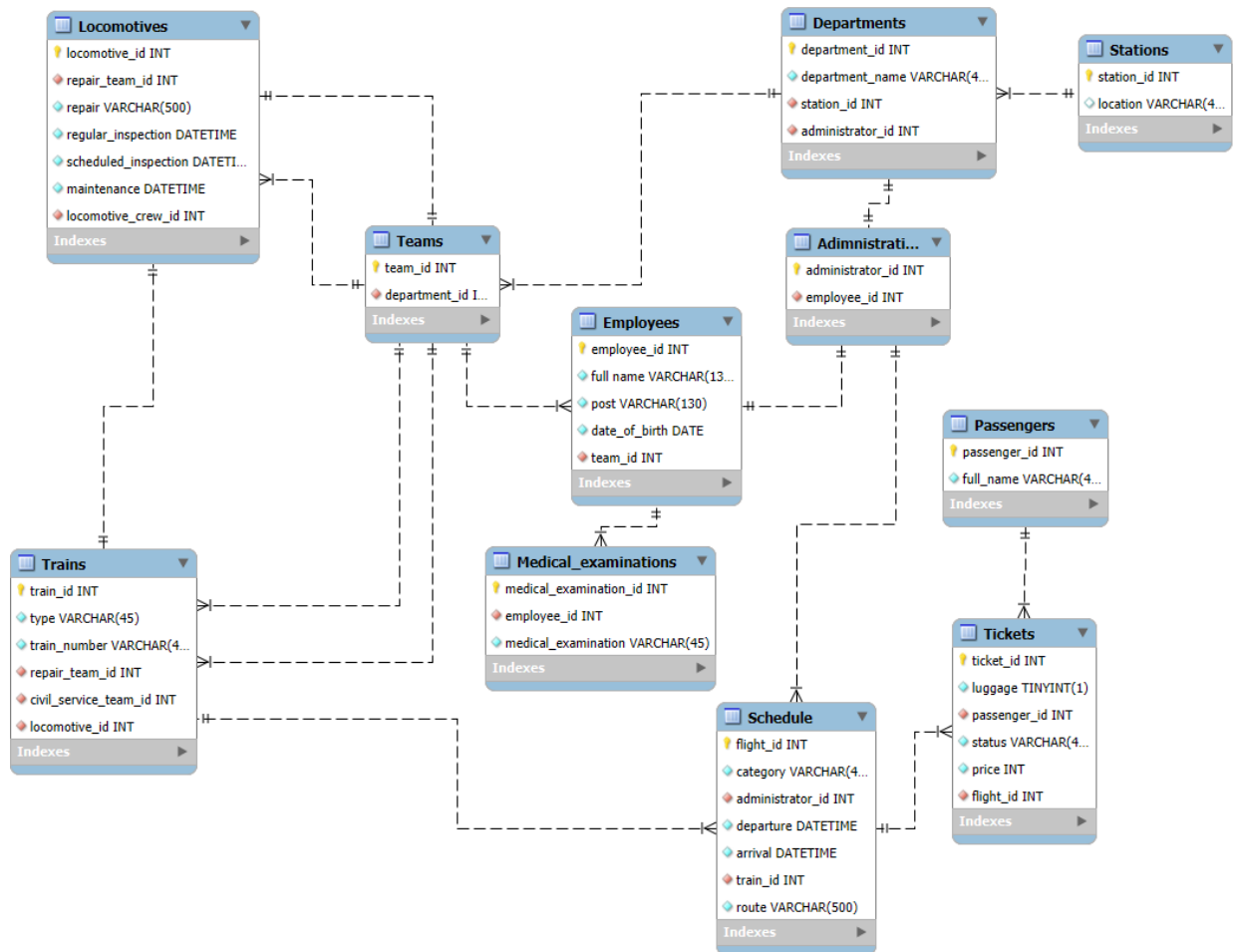


Рисунок 2 – Даталогическая модель

6. Вывод в формате эссе:

Была создана база данных для управления данными железнодорожной станции. Эта база данных включает в себя множество связанных сущностей, которые охватывают все ключевые аспекты функционирования станции. Использование внешних ключей между таблицами, такими как Teams, Employees и Departments, обеспечивает целостность данных, предотвращая наличие «сиротских» записей и гарантируя, что каждая запись имеет соответствующие связи с другими таблицами.

Создание таблиц, таких как Stations, Trains и Schedules, позволяет эффективно управлять информацией о станциях, поездах и расписаниях, что способствует улучшению процессов планирования и работы железнодорожной станции. Структура базы данных достаточно гибкая, что позволяет добавлять новые сущности или атрибуты без значительных

изменений в уже существующих таблицах. Это удобно для будущего расширения функциональности.

Благодаря хорошо продуманной структуре можно легко выполнять запросы для анализа данных, таких как отслеживание работы команд, анализ расписания, учет медицинских обследований сотрудников и так далее. Рекомендуется использовать кодировку `utf8mb4` для поддержки всех символов, включая эмодзи, что делает базу данных более универсальной. Также стоит обратить внимание на использование правильных имен атрибутов, например, в таблице `Employees` поле `full name` можно переименовать в `full_name` для соблюдения единого стиля именования.

В целом, созданная база данных представляет собой эффективный инструмент для управления всеми аспектами работы железнодорожной станции, способствуя повышению эффективности и улучшению обслуживания клиентов.