Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Основы проектирования баз данных

Отчет по практической работе №3

на тему: Проектирование базы данных «Групповые перевозки»

Выполнил: Никитин Семен

Группа: ПР-21

Преподаватель: С.И. Овчинникова

2024

**ВЫПОЛНЕНИЕ РАБОТЫ**

**1.Описание предметной области.**

В данном проекте выполняется разработка реляционной базы данных для информационной системы «Групповые перевозки».

1.1 Постановка задачи.

Главная задача системы – сохранение в базе данных всех необходимых сведений данных о маршруте, информации о водителе, и информации о перевозках. Так как этой базой данных будет пользоваться диспетчера по грузоперевозкам, то внимание было сконцентрировано на том чтобы базой данных было удобно пользоваться и в ней было все интуитивно понятно.

**2.Концептуальная модель базы данных**

При разработке ER-моделей мы должны получить следующую информацию о предметной области:

˗ список сущностей предметной области;

˗ список атрибутов сущностей;

˗ описание взаимосвязей между сущностями.

После анализа предметной области мы выделили три сущности: «Маршруты», «Водители», «Перевозки»». Следующим шагом определим атрибуты всех сущностей и выделим ключевые атрибуты.

***Табл. 1. Атрибуты***

|  |  |  |
| --- | --- | --- |
| Маршруты | Водители | Перевозки |
| IDM  Расстояние  Пункт отправления  Пункт прибытия  Оплата для водителя | IDV  Фамилия  Имя  Отчество  Стаж | IDM  IDV  Дата отправки  Дата прибытия |

Домены, из которых атрибуты берут свои значения, приведены в таблице. Здесь же приведены ограничения для атрибутов на уровне кортежей: повторяемость, обязательность и значения по умолчанию.

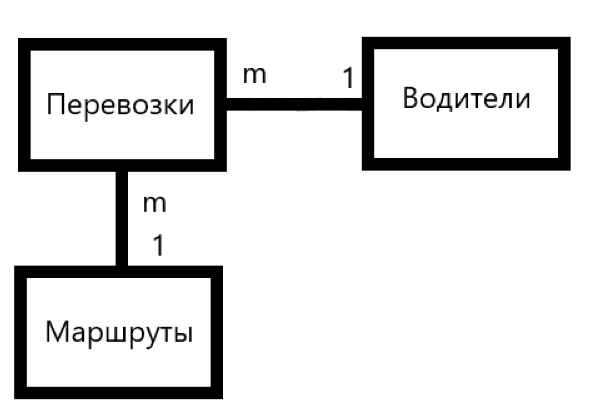
***Таблица 2.*** *Список атрибутов с ограничениями*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Атрибут | Ограничения | Повторяемость | Обязательность | Значение по умолчанию |
| IDM | 1 | - | + | нет |
| Расстояние | 1 | + | + | нет |
| Пункт отправления | 2 | + | + | нет |
| Пункт прибытия | 2 | + | + | нет |
| Оплата для водителя | 1 | + | + | нет |
| IDV | 1 | - | + | нет |
| Фамилия | 2 | + | + | нет |
| Имя | 2 | + | + | нет |
| Отчество | 2 | + | + | нет |
| Стаж | 1 | + | + | нет |
| Дата отправки | 3 | + | + | нет |
| Дата прибытия | 3 | + | + | нет |
| Премия | 1 | + | + | нет |

Примечания:

1. Не может быть отрицательным числом.
2. Строка должна состоять только из символов кириллицы.
3. Только дата.

Определим типы связей и построим начальную ER-модель данных (рис. 1).

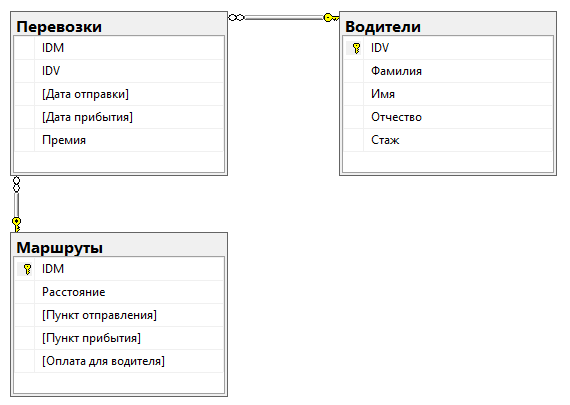


***Рис. 1.*** *Начальная ER - модель данных*

Преобразование концептуальной модели в концептуальную схему выбранной реляционной СУБД осуществляется в следующей последовательности.

* 1. Для каждой сильной сущности ER-модели создается отдельная таблица, а для каждого атрибута сущности создается столбец таблицы. Ключевой атрибут становится первичным ключом, а дополнительные ключевые атрибуты - потенциальными ключами.
  2. Для каждой слабой сущности также создается отдельная таблица, в которой должны присутствовать ключевые столбцы доминирующих таблиц. В зависимости от вида связи устанавливаются ключевые атрибуты таблицы.
  3. Далее необходимо создать внешние ключи, обеспечивающие ссылочную целостность, по указанному типу связи в ER-модели.

Вполне возможно, что в ER-схеме будет присутствовать избыточность данных, поэтому необходимо нормализировать базу данных, как минимум, до нормальной формы Бойса-Кодда.



***Рис. 2.*** *ER-диаграмма*

1. В физической модели каждой сущности будет соответствовать таблица базы данных, а каждому атрибуту – поле таблицы.

***Таблица 3*** *Маршруты*

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| **IDM** | Числовой |
| Расстояние | Числовой |
| Пункт отправления | Символьный |
| Пункт прибытия | Символьный |
| Оплата для водителя | Денежный |

***Таблица 4.*** *Водители*

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| **IDV** | Числовой |
| Фамилия | Символьный |
| Имя | Символьный |
| Отчество | Символьный |
| Стаж | Числовой |

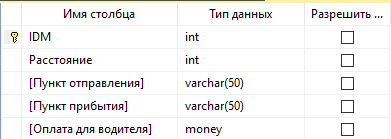
***Таблица 5.*** *Перевозки*

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| IDM | Числовой |
| IDV | Числовой |
| Дата отправки | Дата |
| Дата прибытия | Дата |
| Премия | Денежный |

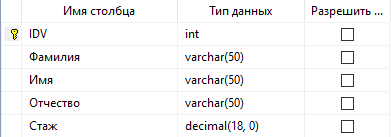
**4.Разработка базы данных**

*4.1. Создание таблиц*

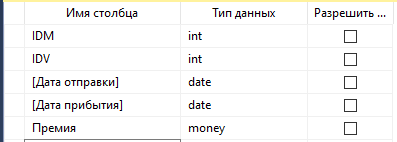
На основании реляционной модели данных необходимо создать 4 таблицы «Маршруты», «Водители», «Перевозки». Создадим таблицы в режиме конструктора. Каждой таблице назначим ключевое поле, кроме таблицы «Перевозки» в нашей базе она слабая сущность (рис. 3,4,5).



***Рис. 3.*** *Таблица «Маршруты» в режиме конструктора*

**

***Рис. 4.*** *Таблица «Водители» в режиме конструктора*

**

***Рис. 5.*** *Таблица «Перевозки» в режиме конструктора*

*4.2. Создание запросов*

Создадим запрос для нахождения заработной платы для водителя, выехавшего из Екатеринбурга. Пример готового результата представлен на рис. 6.

SELECT TOP (1000) [Пункт отправления]

,[Пункт прибытия]

,Фамилия

,Имя

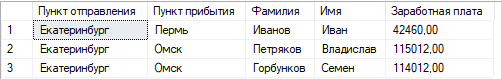
,[Оплата для водителя] + Премия as [Заработная плата]

FROM Маршруты

inner join Перевозки on Маршруты.IDM = Перевозки.IDM

inner join Водители on Водители.IDV = Перевозки.IDV

where [Пункт отправления] = 'Екатеринбург'

******

***Рис 6.*** *Пример выполненного запроса для нахождения заработной платы для водителя, выехавшего из Екатеринбурга*

Создадим запрос для поиска перевозки расстояние которой больше 3000 и стаж водителя больше 6 лет. Выполненный запрос представлен на рис. 7.

SELECT TOP (1000) [Пункт отправления]

,[Пункт прибытия]

,Расстояние

,Фамилия

,Имя

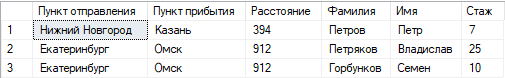
,Стаж

FROM Маршруты

inner join Перевозки on Маршруты.IDM = Перевозки.IDM

inner join Водители on Водители.IDV = Перевозки.IDV

where Расстояние > 3000 and Стаж > 6



***Рис 7****. Пример выполненного запроса для поиска перевозки расстояние которой больше 3000 и стаж водителя больше 6 лет.*

Создадим запрос для определения количества поездок каждого маршрута. Выполненный запрос представлен на рис. 8.

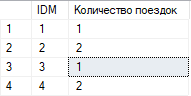
SELECT TOP (1000) Перевозки.IDM

,Count(Перевозки.IDM) as [Количество поездок]

FROM Маршруты

inner join Перевозки on Маршруты.IDM = Перевозки.IDM

group by Перевозки.IDM



***Рис 8****. Пример выполненного запрос на запрос для определения количества поездок каждого маршрута.*

Создадим запрос для поиска перевозки отправленной в 2024 году. Пример готового результата представлен на рис. 9.

SELECT TOP (1000) [Пункт отправления]

,[Пункт прибытия]

,Расстояние

,Фамилия

,Имя

,[Дата отправки]

FROM Маршруты

inner join Перевозки on Маршруты.IDM = Перевозки.IDM

inner join Водители on Водители.IDV = Перевозки.IDV

where Year([Дата отправки]) = 2024



***Рис 9.*** *Пример готового результата для поиска перевозки отправленной в 2024 году*

Вывод: при выполнении работы была разработана база данных «Групповые перевозки» и была проверена с помощью выполнения запросов.