

ИНФОРМАЦИОННАЯ СИСТЕМА ПЛАНИРОВАНИЯ РАБОТЫ ТРАНСПОРТНОЙ КОМПАНИИ

И. С. Гулько, В. О. Ковалева

*ГУО «Институт бизнеса Белорусского государственного
университета», г. Минск;
irinagulko13@gmail.com, valeriakovalyova@mail.ru;
науч. рук. – Е. Г. Гриневич*

В научной работе разработана информационная система на основе клиент-серверной архитектуры для управления работой транспортной компании. Информационная система включает базу данных транспортной компании и приложение для диспетчеров транспортной компании, обеспечивающее планирование и контроль работы водителей. В серверной части приложения, реализованной в СУБД MS SQL Server, разработаны хранимые процедуры, функции и триггеры, обеспечивающие функционал информационной системы. Был разработан простой и понятный интерфейс на языке C# в Windows Forms Application, который позволяет вводить и представлять все необходимые данные для работы диспетчеров транспортной компании. Система может быть внедрена на предприятиях, осуществляющих грузоперевозки.

Ключевые слова: информационная система; автоматизация; клиент-серверная архитектура; база данных; sql; запрос.

Система управления транспортировками, является жизненно важным инструментом в процессе грузоперевозок. Система помогает контролировать все имеющиеся данные о транспортной сети. От вовремя доставленного груза зависит стабильность всего процесса бизнеса. Именно поэтому мы решили разработать базу данных и создать систему, позволяющую планировать работу транспортной компании по перевозке грузов [1, с.105].

Для реализации поставленных целей необходимо решить следующие задачи:

- изучить предметную область;
- разработать структуру программы и базу данных;
- разработать и реализовать клиент-серверное приложение.

В качестве программных средств в работе использованы: ERWin, Visual Studio 2019 с языком Visual C#, Microsoft SQL Server Management Studio.

В ходе работы была спроектирована база данных и создана система для диспетчеров компании грузоперевозок. Система позволяет сохранять данные о: автомобилях, прицепах, сотрудниках, клиентах, контактных лицах, заказах, грузах и расходах водителей. Диспетчеры могут добавлять, изменять и удалять сведения в таблицах. Основная задача состояла в том, что-

бы создать простой и понятный интерфейс, который позволит быстро ввести или получить все необходимые данные для работы.

Информационная система для решения данной проблемы должна поддерживать многопользовательский режим доступа к ней с разных точек. Следовательно, целесообразна ее реализация в системе управления базами данных (СУБД), использующей технологию «клиент-сервер». Для создания серверной части БД выбрана СУБД Microsoft SQL Server 2019, а для клиентской – Visual Studio 2017.

Для начала необходимо было спроектировать физическую структуру таблиц базы данных и связей. Проектирование осуществлялось при помощи инструментальной среды ERwin. ERwin сочетает графический интерфейс Windows, инструменты для построения ER-диаграмм, редакторы для создания логического и физического описания модели данных.

Процесс построения информационной модели состоит из следующих шагов:

- определение сущностей;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы [2, с.47].

Для реализации поставленной цели была спроектирована следующая модель (Рис.4):

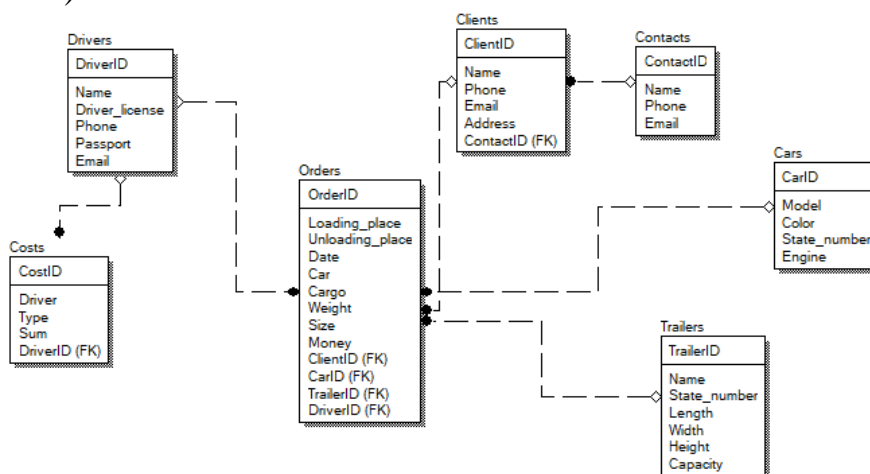


Рис.4 Моделирование IDEF1x – физический уровень

Для хранения всех необходимой информации системы разработана реляционная база данных на платформе MS SQL Server. База данных представлена следующими таблицами: Cars, Trailers, Cleints, Contacts, Drivers, Orders, Costs.

Для реализации клиент-серверной системы написаны хранимые процедуры для работы с таблицами.

На Рис. 5 представлен код запроса на создание функции, которая возвращает список затрат водителей по заданному типу затрат (например, заправка).

```
CREATE FUNCTION fn_SelectionCostsByType
(@Type NCHAR(10))
RETURNS TABLE
AS RETURN
SELECT Costs.Type, Costs.Sum, Drivers.Name
FROM Costs INNER JOIN Drivers
ON Costs.Driver = Drivers.DriverID
WHERE Costs.Type = @Type
GO

SELECT * FROM dbo.fn_SelectionCostsByType('Заправка')
GO
```

Рис. 5 Запрос на создание функции

Был создан триггер, который предотвращает изменения в таблице, путем занесения записи в альтернативную таблицу. На Рис. 6 представлен триггер, который будет добавлять в таблицу «History» данные о добавлении новой машины, изменении информации и удалении машины.

```
CREATE TABLE History
(
    ID INT IDENTITY PRIMARY KEY,
    CarID INT NOT NULL,
    Operation NVARCHAR(200) NOT NULL,
    CreateAt DATETIME NOT NULL DEFAULT GETDATE(),
);

CREATE TRIGGER InsertNewCar
ON Cars
AFTER INSERT
AS
INSERT INTO History (CarID, Operation)
SELECT CarID, 'Добавлена машина ' + Model + 'цвет: ' + Color + 'номер ' + State_number + 'двигатель ' + Engine
FROM INSERTED

INSERT INTO Cars (CarID, Model, Color, State_number, Engine)
VALUES (13, 'Scania', 'Красный', '9112AP-2', '98EEW2')

SELECT * FROM History
```

Рис. 6 Запрос на создание триггера

Интерфейс программы создан с помощью интегрированной среды разработки программного обеспечения Visual Studio 2017.

Главная форма приложения разработана в Windows Forms Applications на языке. C#.

С помощью мастера настройки источника данных добавлена база данных «Transport», её таблицы и хранимые процедуры.

Главная форма приложения представляет собой окно, в верхней части которого расположено меню со всеми основными пунктами, необходимыми для комфортной работы диспетчера компании(Рис.7).



Рис.7 Главная форма приложения

На форме с названием «Cars and Trailers» представлена информация об имеющихся в автопарке компании машинах и прицепах. Имеется возможность добавления, удаления и изменения данных об авто и прицепах. В право части формы – реализация процедуры выбора всех автомобилей той марки, которая была введена пользователем.

На форме «Clients» размещена информация о компаниях-клиентах и контактных лицах этих компаний. Данная форма была реализована с помощью размещения на форме двух связанных таблиц.

Форма «Drivers» включает в себя информацию о водителях и их расходах. Имеется возможность добавления информации о водителях и их расходах, а также удаление. Есть возможность поиска по имени водителя. В правой части – реализация функции, которая возвращает список затрат водителей по заданному типу затрат (например, заправка).

Форма «Orders» содержит информацию об актуальных заказах фирмы. Есть возможность для редактирования, добавления и удаления заказов.

И последняя форма «Reports» предназначена для диспетчеров. Там содержится информация, которая была получена в ходе выполнения триггеров. Таблица «Drivers Cost Data» содержит информацию о действиях с таблицей «Cost». Для этой цели был разработан триггер с курсором. Таблица «Cars Data» содержит информацию, которая была получена в ходе выполнения триггера для таблицы «Cars». Триггер будет добавлять в таблицу «History» данные о добавлении новой машины, изменении информации и удалении машины.

В ходе работы была создана система для планирования и контроля работы водителя дальних перевозок. Создано приложение для диспетчера транспортной компании. Система имеет требуемую функциональности может быть внедрена на предприятиях, осуществляющих транспортные перевозки.

В дальнейшем планируется сопровождение и доработка. В перспективах – создание веб-приложения для водителей, которое будет учитывать личные расходы водителей и связывать его с главным сервером.

Библиографические ссылки

1. *Исаев, Г. Н.* Информационные системы в экономике / Г. Н. Исаев. – Москва: Омега-Л, 2012. – 464 с.
2. *Кузнецов, С. Д.* Основы баз данных / С. Д. Кузнецов. – Москва: Бинوم. Лаборатория знаний, Интернет-университет информационных технологий, 2007. – 484 с.