МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования «БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных технологий

Кафедра Информационные системы и технологии

Специальность 1-40 01 01 «Программное обеспечение информационных

технологий»

Специализация Программирование интернет приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

«База данных «Планирования совместных поездок» с применением технологии шифрования и маскирования в БД»

Выполнил студент Мядель Тимофей Дмитриевич

(Ф.И.О.)

Руководитель проекта пр.-ст. Колмаков Михаил Владимирович

(учен. степень, звание, должность, Ф.И.О., подпись)

Заведующий кафедрой к.т.н., доц. Смелов В.В .

(учен. степень, звание, должность, Ф.И.О., подпись)

Курсовой проект защищен с оценкой

Минск 2021

Оглавление

[Введение 3](#_Toc90892796)

[1. Постановка задачи и анализ прототипов 4](#_Toc90892797)

[2. Разработка базы данных 5](#_Toc90892798)

[2.1 Проектирование базы данных 5](#_Toc90892799)

[2.2 Процедуры для решения поставленных задач 6](#_Toc90892800)

[2.3 Индексы 6](#_Toc90892801)

[2.4 Создание пользователей и ролей 6](#_Toc90892802)

[3. Технология маскирования и шифрования в БД 8](#_Toc90892803)

[4. Импорт и экспорт данных 10](#_Toc90892804)

[5. Тестирование производительности 12](#_Toc90892805)

[Заключение 13](#_Toc90892806)

[Список использованных источников 14](#_Toc90892807)

[Приложение А 15](#_Toc90892808)

[Диаграмма базы данных 15](#_Toc90892809)

[Приложение Б 16](#_Toc90892810)

[Процедуры авторизации и регистрации пользователя 16](#_Toc90892811)

[Процедуры для вывода данных 17](#_Toc90892812)

[Процедуры для покупки билета 19](#_Toc90892813)

[Процедуры для создания 20](#_Toc90892814)

[Процедуры для удаления 23](#_Toc90892815)

[Приложение В 24](#_Toc90892816)

[Процедура экспорта данных в XML 24](#_Toc90892817)

[Приложение Г 25](#_Toc90892818)

# Введение

Целью данной работы была разработка реляционной базы данных на тему «Планирования совместных поездок» с применением технологии шифрования и маскирования в БД.

В связи с тем, что самым важным ресурсом современного мира становится информация, вполне закономерно развитие технологий в направлении хранения и управления данными.

Любое современное web-приложение невозможно представить без базы данных, работающей на основе одной из множества доступных СУБД.

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление, созданием и использованием баз данных.

В настоящее время существует большое количество программ, которые позволяют управлять списком поставленных задач. И данная база данных является гибким и эффективным решением вопросов организаций поездок.

# Постановка задачи и анализ прототипов

На сегодняшний день в ходе развития автопромышленности человек редко передвигается пешком на большие дистанции, однако не все могут или желают водить транспорт из-за чего начали появляться различные сервисы в качестве замены общественному и личному транспорту. В результате развития этих сервисов, передвижение людей на различные дистанции стало очень удобно, легко и экономно.

Начнем рассмотрение аналогов с «BlaBlaCar». Пример его интерфейса представлен на рисунке 1.1.

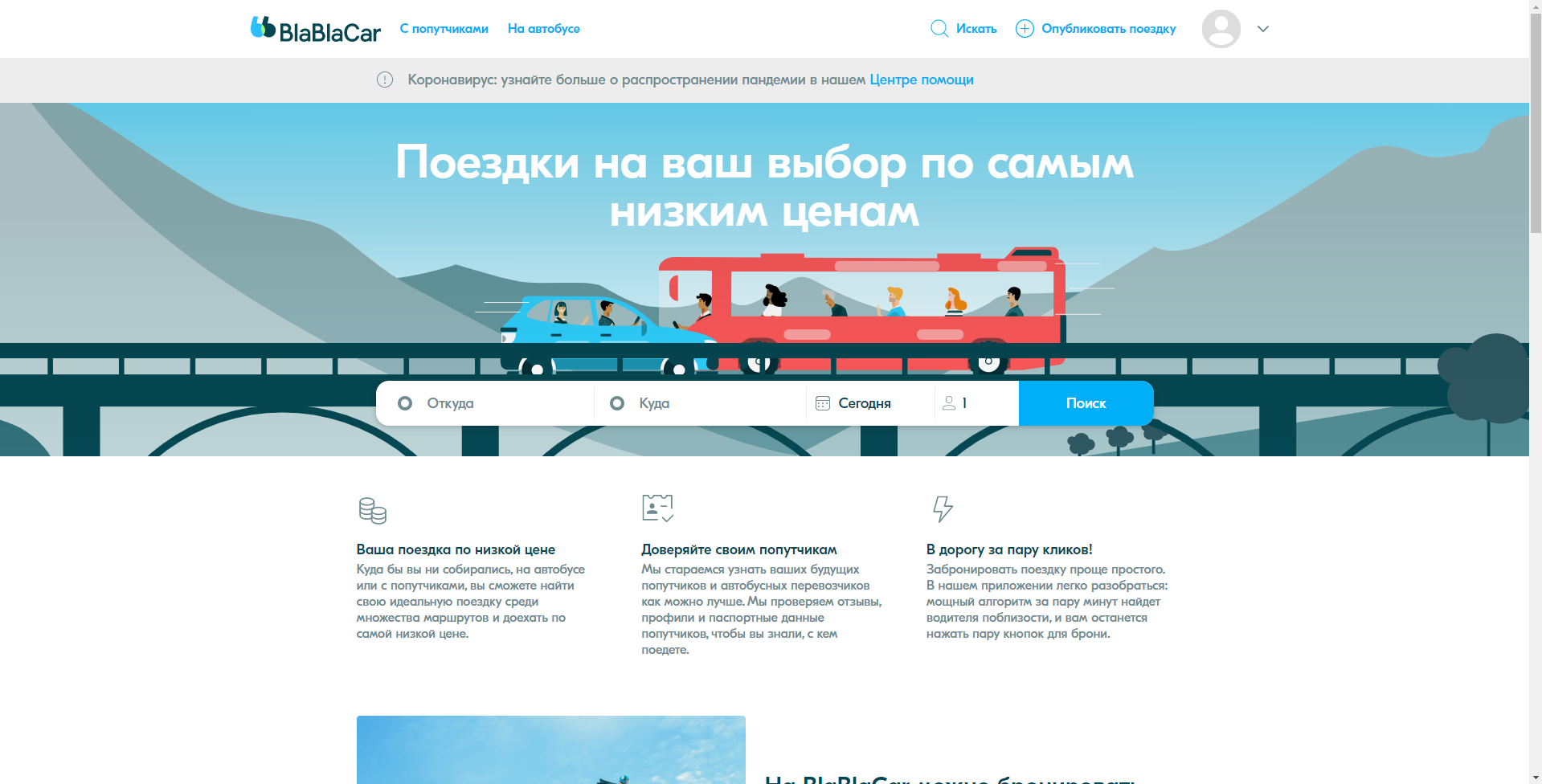


Рисунок 1.1 – Интерфейс сайта популярного сервиса организации поездок «BlaBlaCar»

По итогам изучения аналогов были сделаны выводы, что разрабатываемое приложение должно обладать следующими функциями:

* Создание поездок
* Бронирование мест в поездки
* Просмотр информации о поездке
* Удаление поездок
* Регистрация и авторизация пользователей

# Разработка базы данных

2.1 Проектирование базы данных

При разработке курсового проекта понадобились следующие объекты:

1. Таблицы;
2. Хранимые процедуры;
3. Пользователи
4. Роли
5. Индексы

Для начала рассмотрим таблицы. Их диаграмма представлена на рисунке 2.1, а также в приложении А.

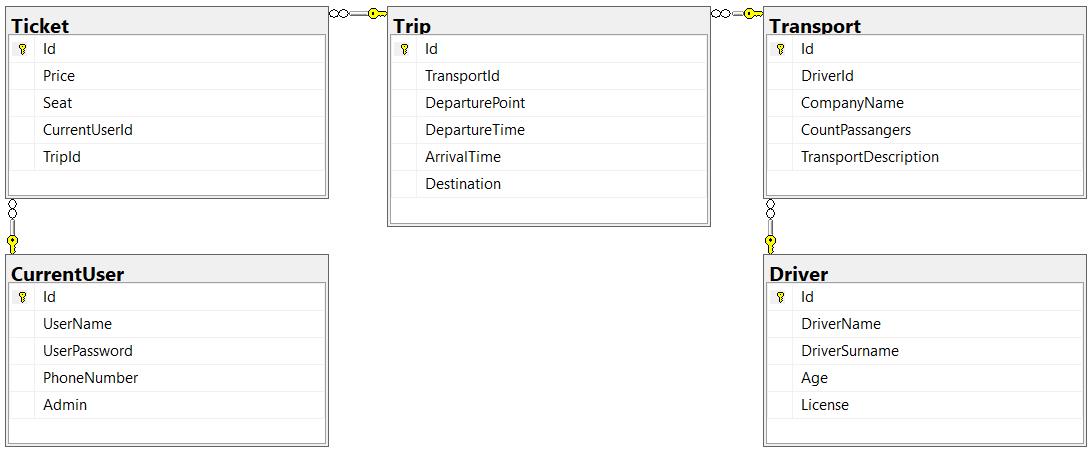


Рисунок 2.1 – Диаграмма таблиц пользовательских данных

База данных приложения состоит из 5 таблиц, связанных между собой внешними ключами.

Таблица «*CurrentUser*» хранит в себе всю информацию о пользователях, включая логин, пароль, номер телефона, а также роль пользователя.

Таблица «*Ticket*» хранит информацию о цене билета, посадочное место, пользователь, который приобрел билет.

Таблица «*Trip*» хранит в себе всю информацию о поездке, точка отправления, время отправления, время прибытия, место назначение.

Таблица «*Transport*» хранит всю информацию о транспорте, фирму, владеющую транспортом, если таковая имеется, количество посадочных мест в транспорте и описание транспорта.

Таблица «*Driver*» содержит информацию о всех водителей, имя, фамилию, возраст, документ, позволяющий управлять транспортным средством.

* 1. Процедуры для решения поставленных задач

Хранимая процедура – объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

При разработке курсового проекта было создано большое количество процедур для следующих целей:

1. Выборка данных из таблиц;
2. Выборка данных по поисковому запросу;
3. Заполнение таблиц 100 000 строк;
4. Вход в аккаунт;
5. Удаление данных из таблиц;
6. Добавление данных в таблицы;
7. Экспорт и импорт таблиц Ticket и Transport в формате xml;

Процедуры представлены в приложении Б.

* 1. Индексы

Для оптимизации работы с поисковыми запросами, необходимо использовать индексы. Так, для работы с наиболее часто используемыми таблицами, были разработаны наборы индексов по наиболее используемым выборкам. Для примера в листинге 2.2 приведено создание индекса indx\_trip\_id.

create index inx\_trip\_id on Ticket(TripId);

Листинг 2.2 – Индекс inx\_trip\_id

* 1. Создание пользователей и ролей

Для базы данных было разработано два типа пользователей. Один из них разработан в качестве пользователя приложения – userdb, а второй для администрирования в базе данных – admindb.

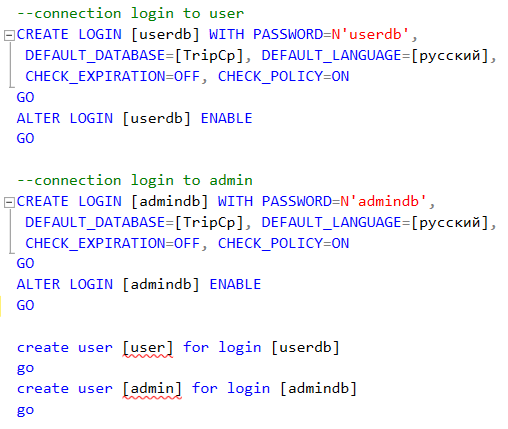


Рисунок 2.2 – Создание пользователей userdb и admindb

# Технология маскирования и шифрования в БД

Динамическое маскирование данных ограничивает возможность раскрытия конфиденциальных данных за счет маскирования этих данных для непривилегированных пользователей. Оно позволяет значительно упростить проектирование и написание кода для системы безопасности в приложении.

Динамическое маскирование данных помогает предотвратить несанкционированный доступ к конфиденциальным данным, позволяя клиентам задать объем раскрываемых конфиденциальных данных с минимальным влиянием на уровень приложения. Динамическое маскирование данных можно настроить для отдельных полей базы данных, чтобы скрыть конфиденциальные данные в результирующих наборах запросов. При использовании динамическое маскирование данных данные в базе данных не изменяются. Динамическое маскирование данных легко использовать с существующими приложениями, так как правила маскирования применяются к результатам запроса. Многие приложения могут маскировать конфиденциальные данные без изменения существующих запросов.

Назначение динамического маскирования данных — ограничение раскрытия конфиденциальных данных, при котором пользователи, у которых нет доступа к данным, не смогут их просматривать. Динамическое маскирование данных не сможет помешать пользователям подключиться к базе данных напрямую и выполнить запросы для получения фрагментов конфиденциальных данных. Динамическое маскирование данных дополняет другие функции безопасности.

Пример маскирования данных представлен на рисунке 3.1.

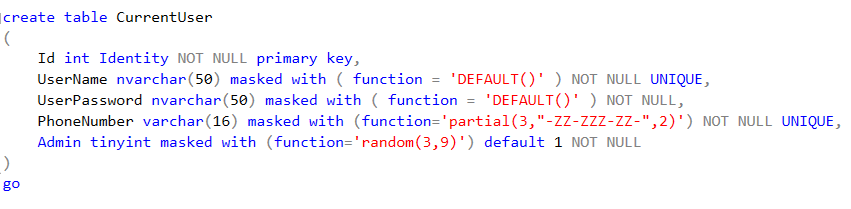


Рисунок 3.1 – Пример маскирования данных

Шифрование представляет собой способ скрытия данных с помощью ключа или пароля. Это делает данные бесполезными без соответствующего ключа или пароля для дешифрования. Шифрование не решает проблемы управления доступом. Однако оно повышает защиту за счет ограничения потери данных даже при обходе системы управления доступом.

Прозрачное шифрование данных позволяет шифровать файлы данных SQL Server.

Функция прозрачного шифрования данных выполняет шифрование и дешифрование ввода-вывода в реальном времени для файлов данных и журналов. Шифрование использует ключ шифрования базы данных (DEK). Загрузочная запись базы данных хранит ключ для доступности во время восстановления. DEK является симметричным ключом. Он защищен сертификатом, который хранится в базе данных master сервера, или асимметричным ключом, который защищает модуль расширенного управления ключами.

Функция прозрачного шифрования данных защищает неактивные данные, то есть файлы данных и журналов. Благодаря ей обеспечивается соответствие требованиям различных законов, постановлений и рекомендаций, действующих в разных отраслях.

Пример включения прозрачного шифрования данных представлен на рисунке 3.1.

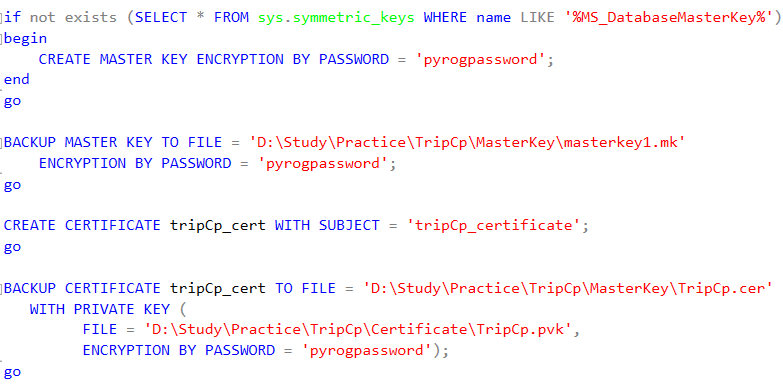


Рисунок 3.1 – Пример включения прозрачного шифрования данных

# Импорт и экспорт данных

Для импорта таблицы Ticket и Transport в формат xml были разработаны процедуры ImportTicket и ImportTransport соответствено. Процедура ImportTransport представлена ниже на рисунке 4.1. Процедура ImportTicket представлена ниже на рисунке 4.2. Отрывок из xml представлен в приложении Г.

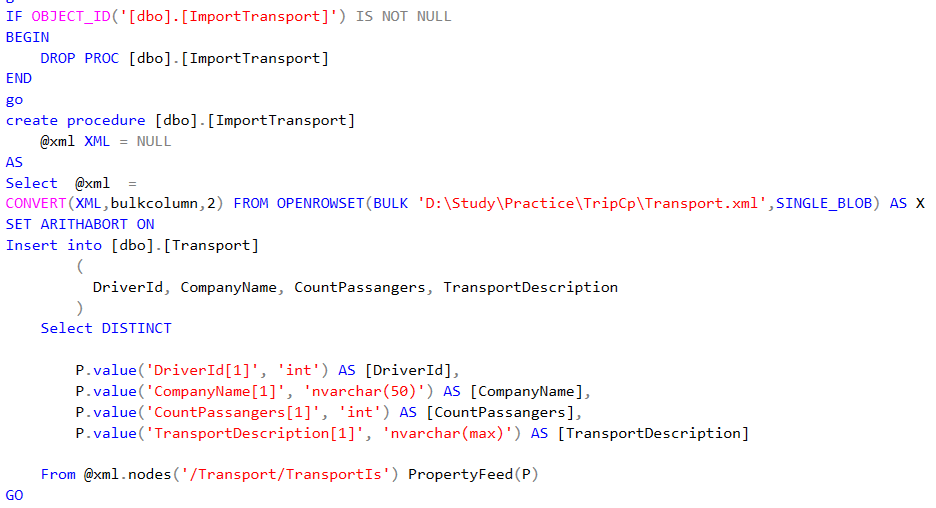


Рисунок 4.1 – Пример создания процедуры ImportTransport

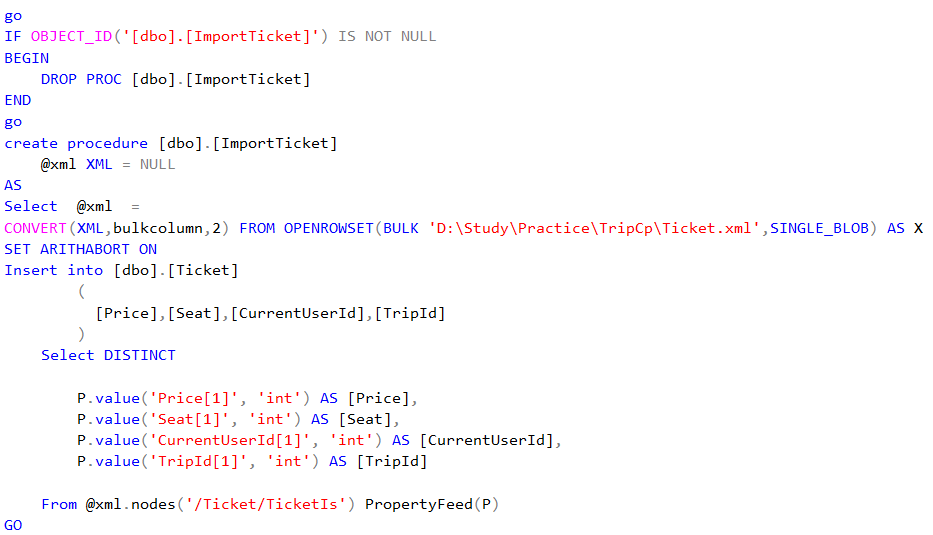


Рисунок 4.2 – Пример создания процедуры ImportTicket

Для экспорта данных в таблицу Transport, в формат xml, была разработана процедура ExportTransport для добавления данных о всех транспортах, зарегистрированных в приложении.

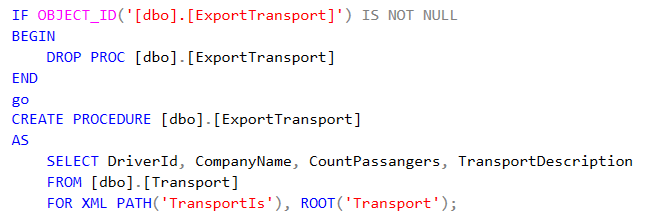


Рисунок 4.2 – Пример создания процедуры ExportTransport

# Тестирование производительности

Для обеспечения тестирования базы данных был проведен запрос для определения стоимости выполнения запроса. Наиболее часто используемой является процедура SelectAllTrip, позволяющая вывести все поездки.

План выполнения и время выполнения процедуры запроса показаны на рисунках 5.1 и 5.2 соответственно.

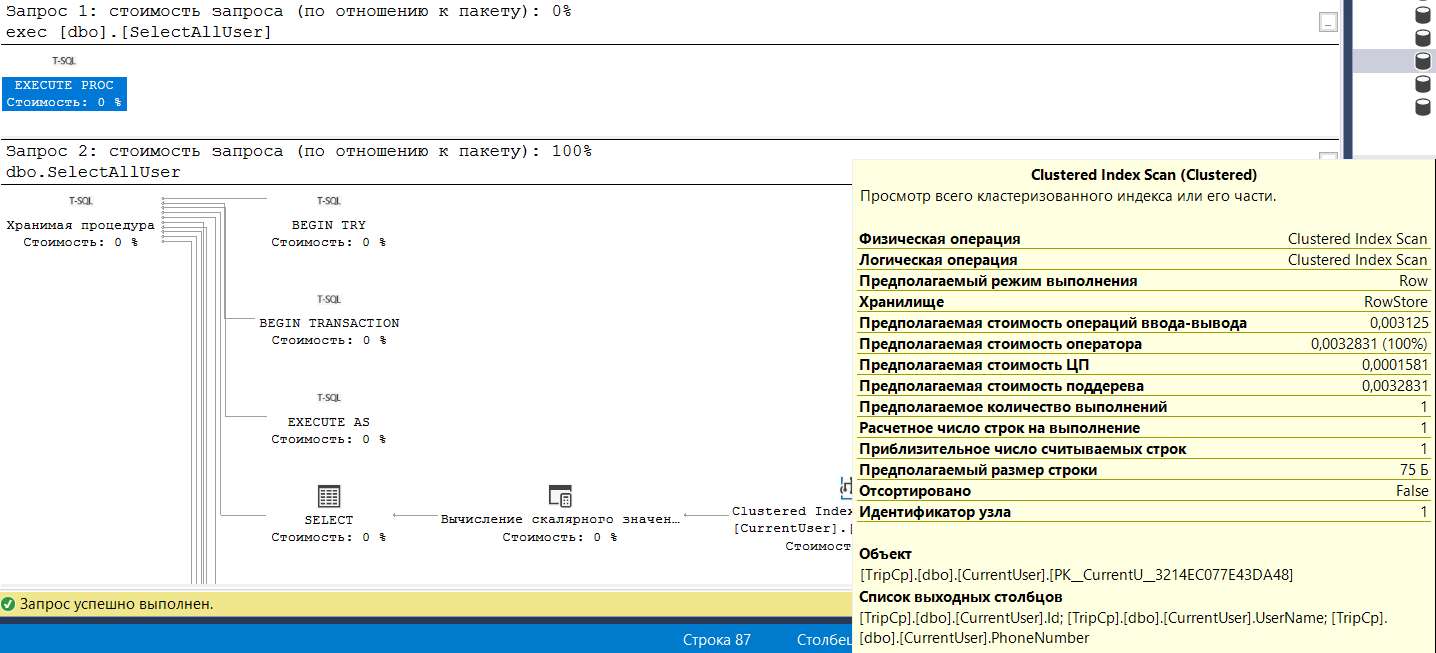


Рисунок 5.1 – План выполнения запроса

Для 1 строки результаты по предполагаемой стоимость операции ввода-вывода запроса – 0,0031, а предполагаемая стоимость ЦП запроса – 0,0032.

Анализируя результаты тестирования, можно прийти к выводу, что приложение будет работать корректно для различного объема данных.

# Заключение

В рамках работы над проектом был проведен обзор аналогичных решений и программных продуктов, выбрана платформа для разработки и система управления базами данных, спроектирована архитектура и структура базы данных, проведено тестирование производительности запросов к таблицам базы данных при различном объеме данных.

Одной из главных задач данной курсовой работы - освоение технологии шифрование и маскирование данных. Данная технология и ее возможности были успешно изучены и реализованы в процессе разработки базы данных.

Данная база данных соответствует представленной задаче и отвечает всем необходимым требованиям:

* Создание поездок
* Бронирование мест в поездки
* Просмотр информации о поездке
* Удаление поездок
* Регистрация и авторизация пользователей

Разработанная база банных содержит в себе 5 связанных таблиц. Разработка базы данных проводилась в рамках MS SQL Server – система управления реляционными базами данных. Данная СУБД используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

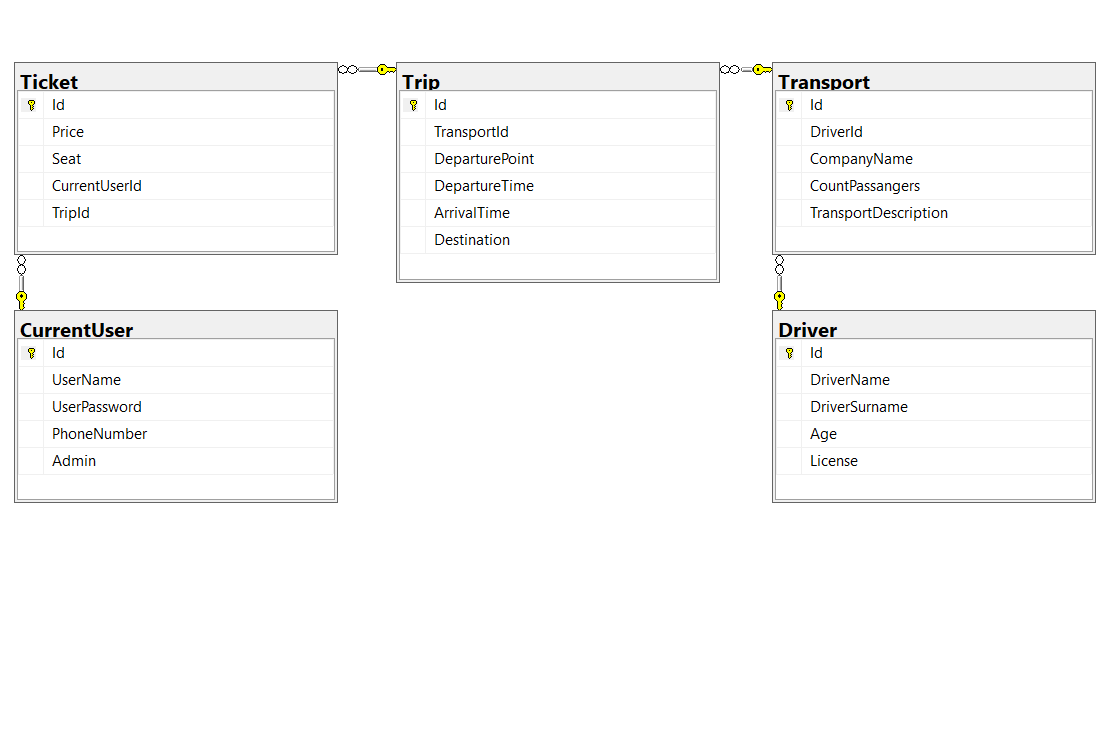
База данных законченной, хотя и возможна доработка и расширение функционала.

# Список использованных источников

1. Transparent Data Encryption (TDE) SSMS [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/relational-databases/security/encryption/transparent-data-encryption?view=sql-server-ver15>
2. Dynamic Data Masking (DDM) [Электронный ресурс] / Microsoft. – Режим доступа: <https://docs.microsoft.com/en-us/sql/relational-databases/security/dynamic-data-masking?view=sql-server-ver15>
3. Техническая документация по SQL Server / Microsoft. – Режим доступа: <https://docs.microsoft.com/ru-ru/sql/sql-server/?view=sql-server-ver15>

# Приложение А

Диаграмма базы данных



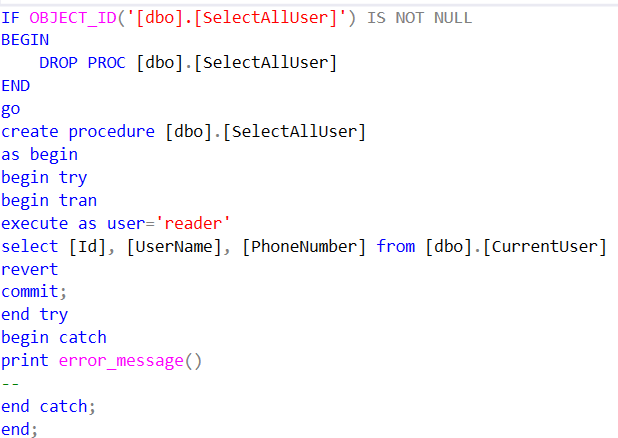
# Приложение Б

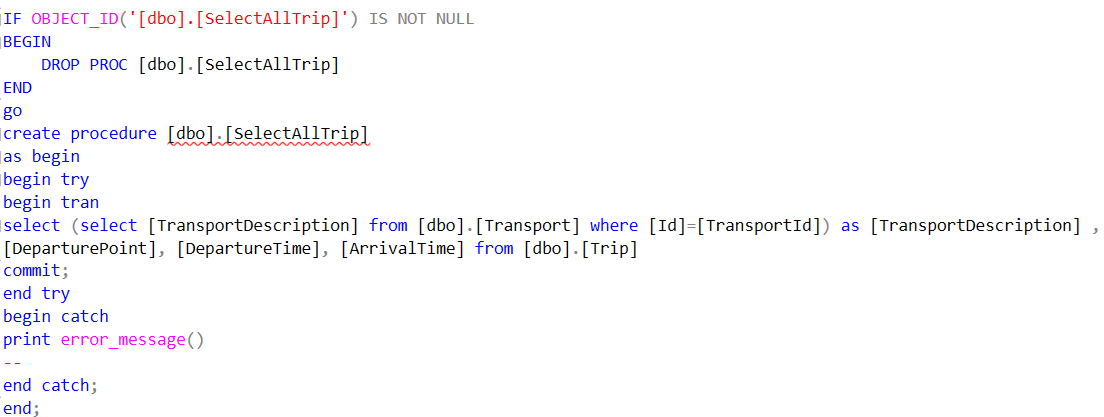
Процедуры авторизации и регистрации пользователя

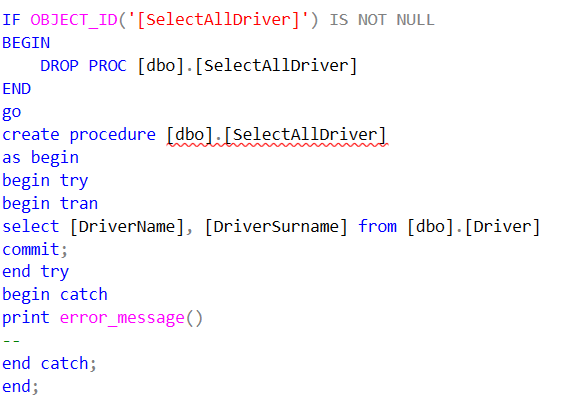


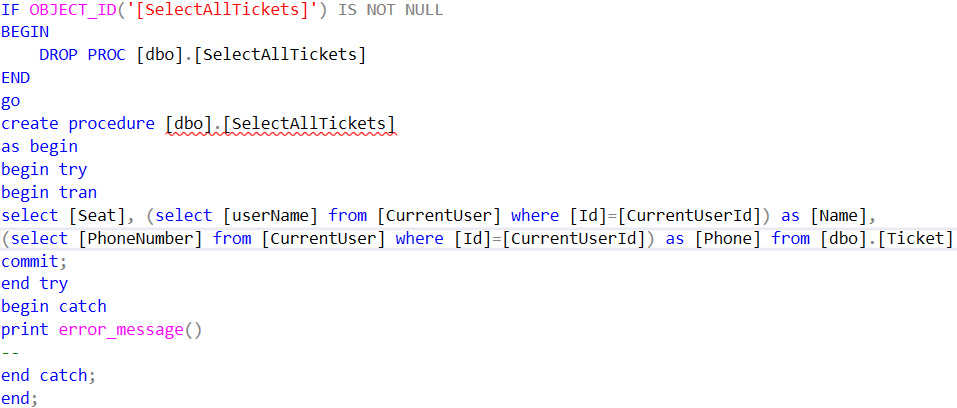


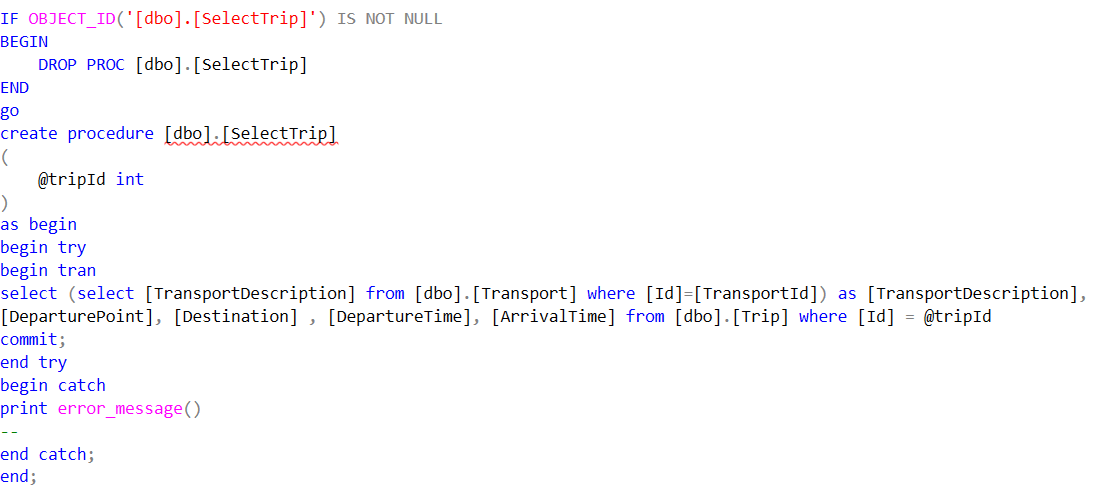
Процедуры для вывода данных



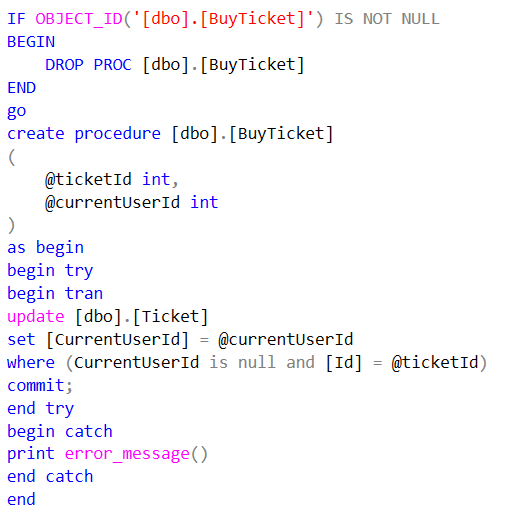




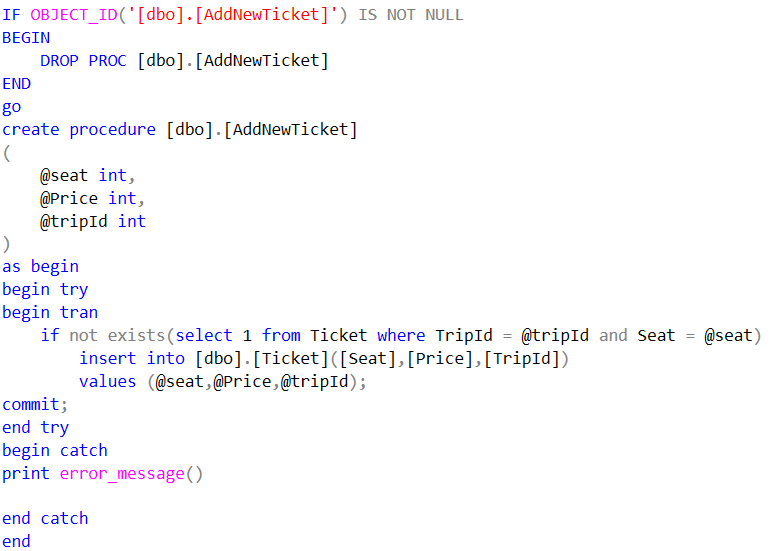


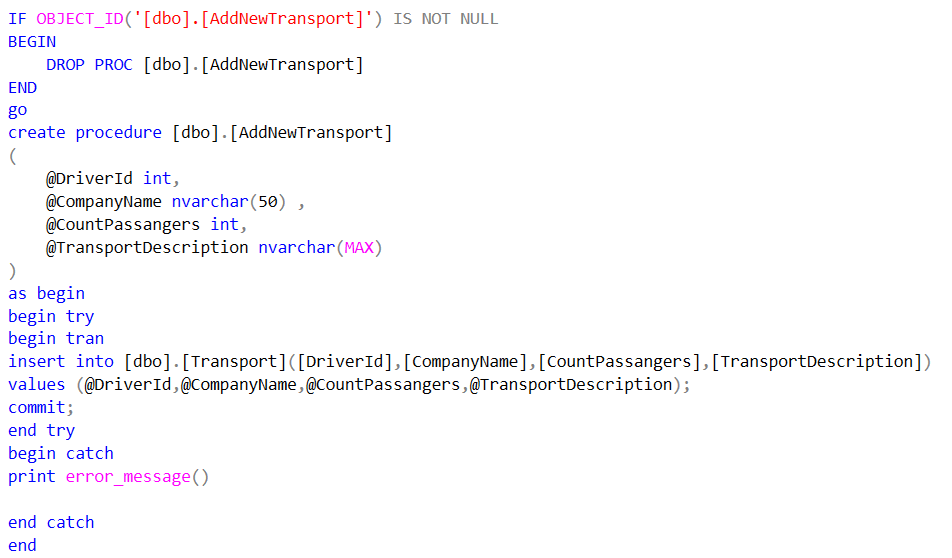


Процедуры для покупки билета

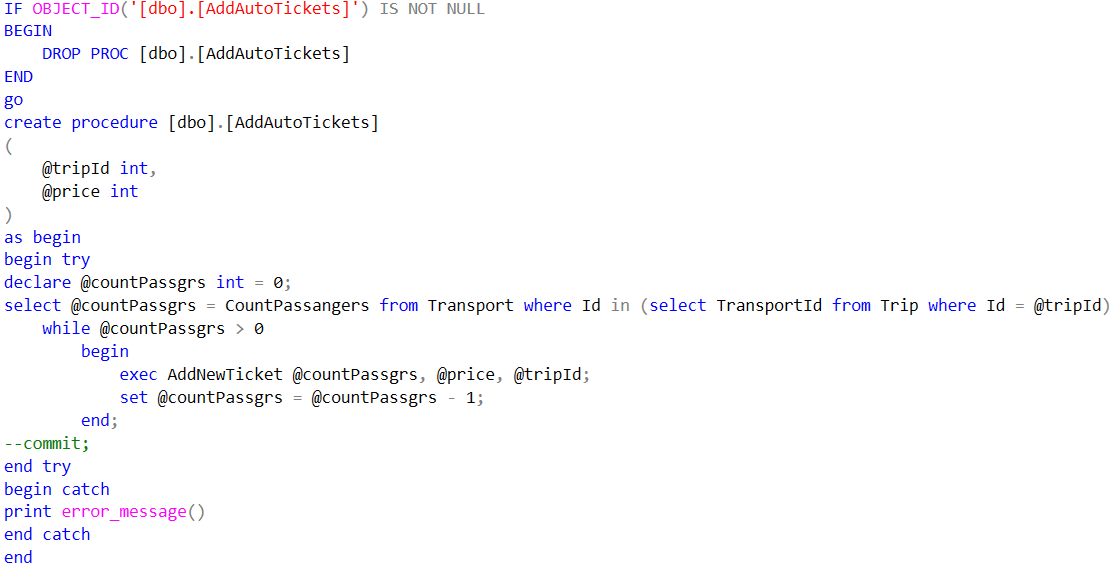


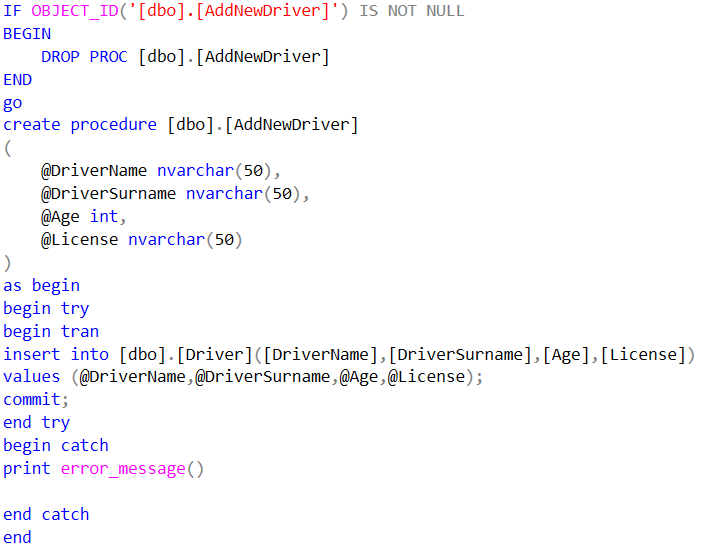
Процедуры для создания



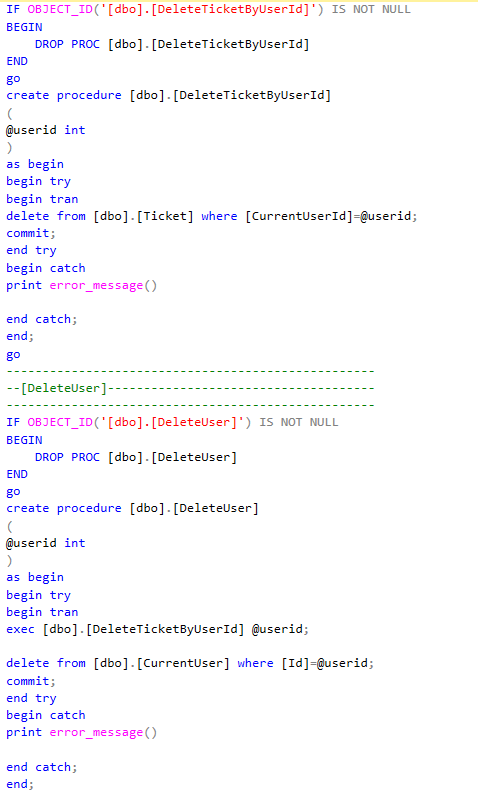






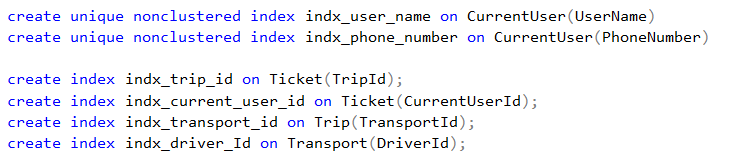


Процедуры для удаления



# Приложение В

Процедура экспорта данных в XML



# Приложение Г

