БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

**ИТОГОВЫЙ ОТЧЕТ**

учебной дисциплины

«Модели данных и системы управления базами данных»

**Выполнил:**

Горбач Владимир Игоревич

**Преподаватель:**

Дрепакова Ангелина Валерьевна

Минск, 2024

Содержание

[Лабораторная работа 2 4](#_Toc167118904)

[Постановка задачи 4](#_Toc167118905)

[1. Анализ предметной области 4](#_Toc167118906)

[1.1 Сущность “Депо” 5](#_Toc167118907)

[1.2 Сущность “Транспорт” 5](#_Toc167118908)

[1.3 Сущность “Водитель” 5](#_Toc167118909)

[1.4 Сущность “Маршрут” 5](#_Toc167118910)

[1.5 Сущность “Остановка” 6](#_Toc167118911)

[2. Построение концептуальной модели 7](#_Toc167118912)

[3. Логическое проектирование реляционной БД 8](#_Toc167118913)

[4. Составление и нормализация реляционных отношений 9](#_Toc167118914)

[4.1 Схема отношения «Депо» (Depot): 9](#_Toc167118915)

[4.2 Схема отношения «Транспорт» (Transport): 9](#_Toc167118916)

[4.3 Схема отношения «Маршрут» (Route): 9](#_Toc167118917)

[4.4 Схема отношения «Остановка» (Stop): 9](#_Toc167118918)

[4.5 Схема отношения «Водитель» (Driver): 10](#_Toc167118919)

[4.6 Схема отношения «Посредник между Депо и Транспортом» (Depot\_Transport): 10](#_Toc167118920)

[4.6 Схема отношения «Посредник между Маршрутом и Остановкой» (Route\_Stop): 10](#_Toc167118921)

[Лабораторная работа 2-4 11](#_Toc167118922)

[Задание 1 11](#_Toc167118923)

[Задание 2 12](#_Toc167118924)

[Задание 3 13](#_Toc167118925)

[Задание 4 16](#_Toc167118926)

[Задание 5 18](#_Toc167118927)

[Задание 6 19](#_Toc167118928)

[Задание 7 21](#_Toc167118929)

[Задание 8 22](#_Toc167118930)

[Задание 9 23](#_Toc167118931)

[Задание 10 24](#_Toc167118932)

[Задание 11 25](#_Toc167118933)

[Задание 12 27](#_Toc167118934)

[Лабораторная работа 5-6 28](#_Toc167118935)

[Задание 13 28](#_Toc167118936)

[Задание 14 35](#_Toc167118937)

[Задание 15 38](#_Toc167118938)

[Лабораторная работа 7-9 40](#_Toc167118939)

[Задание 16 40](#_Toc167118940)

[Задание 17 42](#_Toc167118941)

[Задание 18 44](#_Toc167118942)

[Задание 19 48](#_Toc167118943)

[Задание 20 52](#_Toc167118944)

[Задание 21 53](#_Toc167118945)

[Задание 23 54](#_Toc167118946)

# Лабораторная работа 2

## Постановка задачи

Система управления общественным транспортом предоставляет информацию о его организационной структуре, включая депо, различные виды транспорта, водителей, маршруты и остановки. Каждое депо имеет уникальный идентификатор (ID) и название. Виды транспорта привязаны к определенным депо и содержат информацию об уникальном идентификаторе, названии и уникальном идентификаторе депо. Маршруты обладают уникальным идентификатором, названием и информацией о расписании движения. Остановки также имеют уникальный идентификатор, название и координаты на карте.

Задача включает в себя возможность отслеживания состояния и местоположения транспортных средств, учета водителей и пассажиров, а также планирования и контроля маршрутов.

## 1. Анализ предметной области

В данной предметной области система управления общественным транспортом представляет собой основную структуру, охватывающую депо, различные виды транспорта, водителей, маршруты и остановки. Основная цель системы - обеспечение эффективной работы общественного транспорта и учет информации о водителях, маршрутах и остановках. Анализируя предметную область, можно выделить следующие требования к модели:

• Депо:

Необходимо учитывать информацию о депо, включая уникальные идентификаторы и названия.

• Виды транспорта:

Для каждого вида транспорта важны уникальные идентификаторы, названия и связь с соответствующим депо.

• Маршруты:

Требуется учет информации о маршрутах, включая уникальные идентификаторы и названия.

• Остановки:

Важна информация об остановках, такая как уникальные идентификаторы, названия и координаты на карте.

• Водители:

Для водителей важны уникальные идентификаторы, информация о персональных данных и связь с определенным видом транспорта.

В результате анализа предметной области было выделено пять сущностей:

### 1.1 Сущность “Депо”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание** | **Тип, домен** |
| ID | Уникальный номер, ключевой атрибут. | Целое число |
| Номер | Гос. номер депо | Целое число |

### 1.2 Сущность “Транспорт”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание** | **Тип, домен** |
| ID | Уникальный номер, ключевой атрибут. | Целое число |
| ID маршрута | Уникальный номер маршрута, прикрепленного к транспортному средству | Целое число |
| Тип | Тип транспорта (автобус, троллейбус и т.д.) | Перечисление |
| Гос. номер | Государственный номер транспортного средства | Текст |
| Марка | Марка транспортного средства | Текст |
| Вместимость | Количество пассажиров | Целое число |
| Номер автобуса | Номер транспортного средства в транспортной системе | Целое число |

### 1.3 Сущность “Водитель”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание** | **Тип, домен** |
| ID | Уникальный номер, ключевой атрибут. | Целое число |
| ID транспорта | Уникальный номер транспорта, прикреплённого к водителю | Целое число |
| Имя | ФИО водителя | Текст |
| Телефон | Номер телефона водителя | Текст |
| Категории | Водительские категории | Текст |

### 1.4 Сущность “Маршрут”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание** | **Тип, домен** |
| ID | Уникальный номер, ключевой атрибут. | Целое число |
| Номер | Номер маршрута | Целое число |

### 1.5 Сущность “Остановка”

|  |  |  |
| --- | --- | --- |
| **Атрибут** | **Описание** | **Тип, домен** |
| ID | Уникальный номер, ключевой атрибут. | Целое число |
| Название | Название остановки | Текст |
| Х | Х координата остановки | Целое число |
| Y | Y координата остановки | Целое число |

## 2. Построение концептуальной модели

В данной предметной области сущности взаимосвязаны следующим образом, что требует отражения в базе данных:

В данной предметной области сущности взаимосвязаны следующим образом, что требует отражения в базе данных:

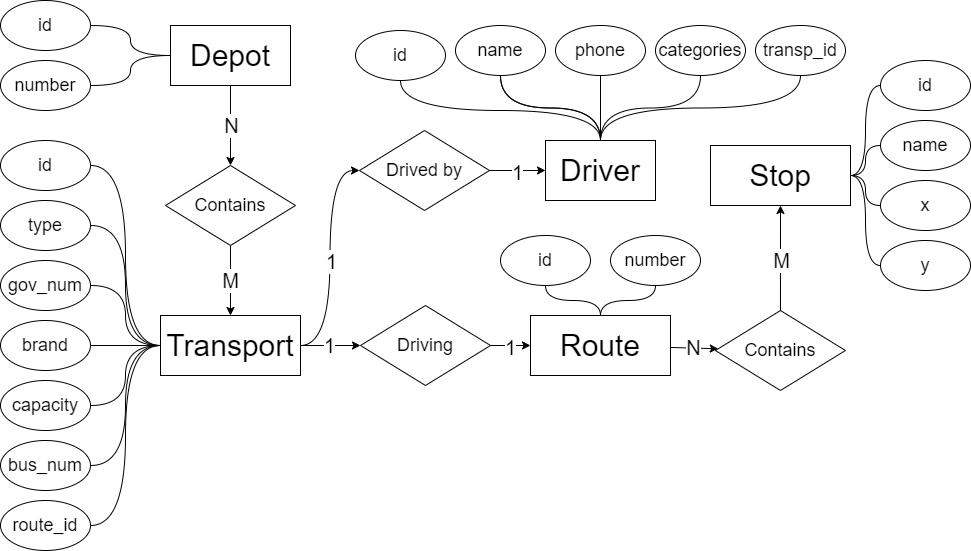
Депо - Транспорт: устанавливается отношение "один ко многим" между депо и видами транспорта. Каждое депо может иметь много видов транспорта, но каждый вид транспорта принадлежит только одному депо. Это отражено через внешний ключ в таблице "Транспорт", который связан с первичным ключом в таблице "Депо".

Транспорт - Водитель: устанавливается отношение "один ко многим" между видом транспорта и водителями. Каждый вид транспорта может иметь много водителей, но каждый водитель связан только с одним видом транспорта. Это также отражено через внешний ключ в таблице "Водитель", связанный с первичным ключом в таблице "Транспорт".

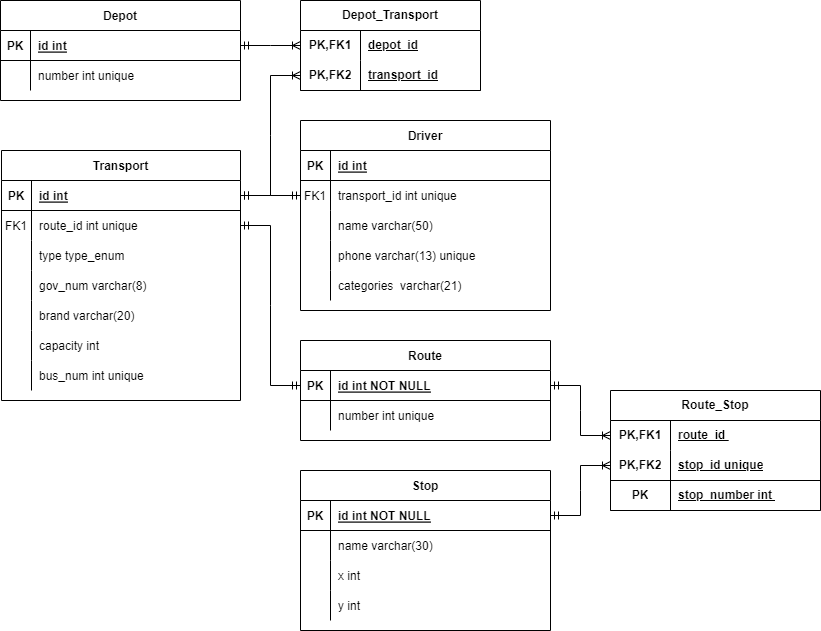
Транспорт - Маршрут: устанавливается отношение "один ко многим" между видом транспорта и маршрутами. Каждый вид транспорта может иметь много маршрутов, но каждый маршрут принадлежит только одному виду транспорта. Это реализуется через внешний ключ в таблице "Маршрут", связанный с первичным ключом в таблице "Транспорт".

Маршрут - Остановка: устанавливается отношение "многие ко многим" между маршрутами и остановками. Каждый маршрут может проходить через много остановок, и каждая остановка может быть частью множества маршрутов. Для отражения этой связи создается дополнительная таблица "Маршрут\_Остановка", которая содержит внешние ключи к таблицам "Маршрут" и "Остановка".

Таким образом, база данных должна содержать соответствующие таблицы для каждой сущности, а также связи между ними через внешние ключи, чтобы отразить указанные отношения.



## 3. Логическое проектирование реляционной БД



## 4. Составление и нормализация реляционных отношений

### 4.1 Схема отношения «Депо» (Depot):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| id | int | 4 | IDENTITY(1,1) | обязательное поле | Первичный ключ |
| number | int | 4 | **-** | обязательное поле | **-** |

### 4.2 Схема отношения «Транспорт» (Transport):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| id | int | 4 | IDENTITY(1,1) | обязательное поле | Первичный ключ |
| route\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Route. |
| type | enum | 4 | **-** | обязательное поле | **-** |
| gov\_num | varchar | 8 | **-** | обязательное поле | **-** |
| brand | varchar | 20 | **-** | обязательное поле | **-** |
| cappacity | int | 4 | **-** | обязательное поле | **-** |
| bus\_num | int | 4 | **-** | обязательное поле | **-** |

### 4.3 Схема отношения «Маршрут» (Route):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| id | int | 4 | IDENTITY(1,1) | обязательное поле | Первичный ключ |
| number | int | 4 | **-** | обязательное поле | **-** |

### 4.4 Схема отношения «Остановка» (Stop):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| id | int | 4 | IDENTITY(1,1) | обязательное поле | Первичный ключ |
| name | varchar | 30 | **-** | обязательное поле | **-** |
| x | int | 4 | **-** | обязательное поле | **-** |
| y | int | 4 | **-** | обязательное поле | **-** |

### 4.5 Схема отношения «Водитель» (Driver):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| id | int | 4 | IDENTITY(1,1) | обязательное поле | Первичный ключ |
| transport\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Transport. |
| name | varchar | 50 | **-** | обязательное поле | **-** |
| phone | varchar | 13 | **-** | обязательное поле | **-** |
| categories | varchar | 21 | **-** | обязательное поле | **-** |

### 4.6 Схема отношения «Посредник между Депо и Транспортом» (Depot\_Transport):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| depot\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Depot |
| transport\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Transport |

### 4.6 Схема отношения «Посредник между Маршрутом и Остановкой» (Route\_Stop):

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Наименование поля** | **Тип поля** | **Размер поля** | **Значение по умолчанию** | **Ограничения** | **Ключ или индекс** |
| route\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Route |
| stop\_id | int | 4 | **-** | обязательное поле | Внешний ключ к Stop |
| stop\_number | int | 4 | **-** | обязательное поле | Первичный ключ |

Отношения были изначально разработаны с учетом требований нормальных форм, что обеспечивает их соответствие третьей нормальной форме (3НФ). Это подразумевает следующие характеристики:

* Неключевые столбцы зависят от всего первичного ключа, а не от его отдельных частей.
* Нет множественных столбцов, представляющих атомарные значения
* Нет повторяющихся столбцов в отношениях.
* Определен первичный ключ для каждого отношения.

Неключевые столбцы зависят исключительно от первичного ключа, и нет транзитивных зависимостей от других неключевых столбцов.

# Лабораторная работа 2-4

## Задание 1

Постановка задачи: Система управления общественным транспортом предоставляет информацию о его организационной структуре, включая депо, различные виды транспорта, водителей, маршруты и остановки. Каждое депо имеет уникальный идентификатор (ID) и название. Виды транспорта привязаны к определенным депо и содержат информацию об уникальном идентификаторе, названии и уникальном идентификаторе депо. Маршруты обладают уникальным идентификатором, названием и информацией о расписании движения. Остановки также имеют уникальный идентификатор, название и координаты на карте.

Описание предметной области: В данной предметной области система управления общественным транспортом представляет собой основную структуру, охватывающую депо, различные виды транспорта, водителей, маршруты и остановки. Основная цель системы - обеспечение эффективной работы общественного транспорта и учет информации о водителях, маршрутах и остановках. Анализируя предметную область, можно выделить следующие требования к модели:

• Депо:

Необходимо учитывать информацию о депо, включая уникальные идентификаторы и названия.

• Виды транспорта:

Для каждого вида транспорта важны уникальные идентификаторы, названия и связь с соответствующим депо.

• Маршруты:

Требуется учет информации о маршрутах, включая уникальные идентификаторы и названия.

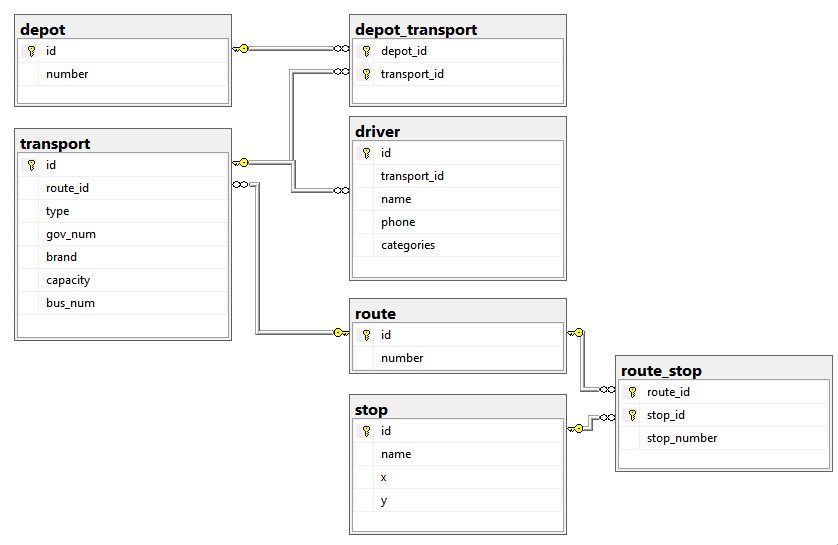
• Остановки:

Важна информация об остановках, такая как уникальные идентификаторы, названия и координаты на карте.

• Водители:

Для водителей важны уникальные идентификаторы, информация о персональных данных и связь с определенным видом транспорта.

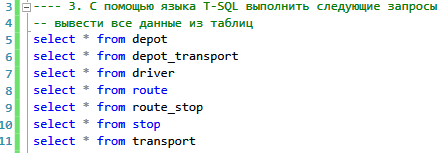
## Задание 2



## Задание 3

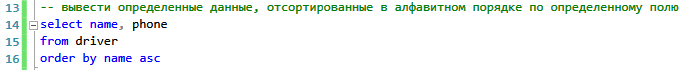
Очень рад, что получил еще одну попытку помочь вам! Вот подробные пояснения для каждого запроса на языке T-SQL:

1. Вывести все данные из таблиц "depot", "depot\_transport", "driver", "route", "route\_stop", "stop" и "transport"

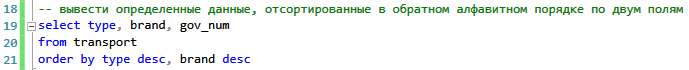


Этот запрос выбирает все строки и все столбцы из таблицы "depot", "depot\_transport", "driver", "route", "route\_stop", "stop" и "transport" и выводит их на экран.

2. Вывести определенные данные из таблицы "driver" - имена и телефоны, отсортированные в алфавитном порядке по имени:

 Этот запрос выбирает столбцы "name" и "phone" из таблицы "driver". Результаты сортируются в алфавитном порядке по столбцу "name" в порядке возрастания (ASC).

3. Вывести определенные данные из таблицы "transport" - тип, марку и государственный номер, отсортированные в обратном алфавитном порядке по типу, а затем по марке:



Этот запрос выбирает столбцы "type", "brand" и "gov\_num" из таблицы "transport". Результаты сортируются в обратном алфавитном порядке по столбцу "type" в порядке убывания (DESC), а затем по столбцу "brand" также в порядке убывания.

4. Создать вычисляемое поле "cat\_num", представляющее длину значения столбца "categories", и вывести уникальные значения столбца "categories" и соответствующие значения "cat\_num" без дубликатов:



Этот запрос выбирает уникальные значения столбца "categories" из таблицы "driver". Для каждого значения столбца "categories" вычисляется его длина с помощью функции LEN(). Результаты выводятся в столбцах "categories" и "cat\_num".

5. Вывести 30% строк из таблицы "stop":



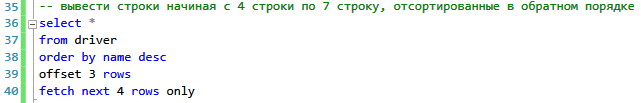
Этот запрос выбирает верхние 30% строк из таблицы "stop" и выводит все столбцы этих строк на экран.

6. Вывести первые 5 строк из таблицы "driver":

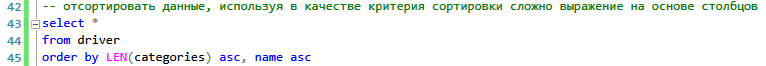


Этот запрос выбирает верхние 5 строк из таблицы "driver" и выводит все столбцы этих строк на экран.

7. Вывести строки, начиная с 4 строки и до 7 строки, отсортированные в обратном порядке по имени:

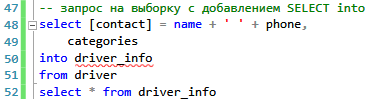
Этот запрос выбирает строки с 4 по 7 из таблицы "driver". Результаты сортируются в обратном порядке по столбцу "name" и пропускаются первые 3 строки с помощью OFFSET, затем выбираются следующие 4 строки с помощью FETCH NEXT.

8. Отсортировать данные, используя в качестве критерия сортировки сложное выражение на основе столбцов



В этом запросе выбираются все столбцы из таблицы "driver". Результаты сортируются в порядке возрастания длины значения в столбце "categories" с помощью выражения "ORDER BY LEN(categories) ASC". Затем результаты сортируются по алфавиту по столбцу "name" с помощью выражения "ORDER BY name ASC". Таким образом, данные сначала сортируются по возрастанию длины значения в столбце "categories", а затем по алфавиту по столбцу "name".

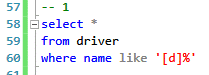
9. Запрос на выборку с добавлением SELECT INTO



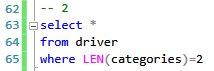
В этом запросе выбираются столбцы "name" и "phone" из таблицы "driver". Каждая выбранная запись объединяется в одно значение в столбце "contact", где имя и номер телефона разделены пробелом. Затем выбранные столбцы "contact" и "categories" вставляются в новую таблицу "driver\_info" с помощью оператора SELECT INTO. И наконец, результаты из новой таблицы "driver\_info" выводятся.

## Задание 4

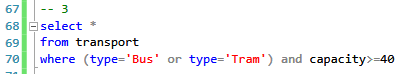
1. В этом запросе выбираются все столбцы из таблицы "driver", где значение в столбце "name" начинается с буквы "d" (используется шаблон "[d]%").



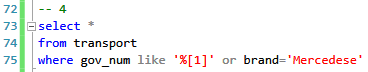
2. В этом запросе выбираются все столбцы из таблицы "driver", где длина значения в столбце "categories" равна 2.



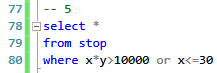
3. В этом запросе выбираются все столбцы из таблицы "transport", где значение в столбце "type" равно "Bus" или "Tram", а значение в столбце "capacity" больше или равно 40.



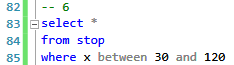
4. В этом запросе выбираются все столбцы из таблицы "transport", где значение в столбце "gov\_num" оканчивается на цифру "1" или значение в столбце "brand" равно "Mercedes".



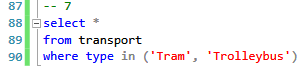
5. В этом запросе выбираются все столбцы из таблицы "stop", где произведение значений в столбцах "x" и "y" больше 10000 или значение в столбце "x" меньше или равно 30.



6. В этом запросе выбираются все столбцы из таблицы "stop", где значение в столбце "x" находится в диапазоне между 30 и 120 (включительно).



7. В этом запросе выбираются все столбцы из таблицы "transport", где значение в столбце "type" равно "Tram" или "Trolleybus".



## Задание 5

Переименовать одну из таблиц вашей БД

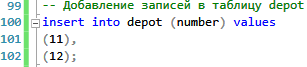


В этой команде выполняется переименование таблицы "driver\_info" в "driver\_information" с помощью хранимой процедуры sp\_rename. Первый параметр указывает текущее имя таблицы, а второй параметр указывает новое имя таблицы.

## Задание 6

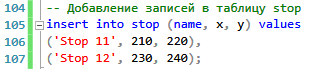
Добавление записей в каждую из таблиц вашей БД

1. Добавление записей в таблицу depot



В этом запросе выполняется добавление двух записей в таблицу "depot". В столбец "number" вставляются значения 11 и 12.

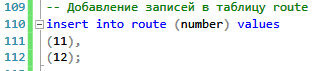
1. Добавление записей в таблицу stop



В этом запросе выполняется добавление двух записей в таблицу "stop". В столбцы "name", "x" и "y" вставляются следующие значения:

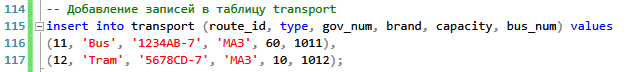
* 'Stop 11', 210, 220
* 'Stop 12', 230, 240

1. Добавление записей в таблицу route



В этом запросе выполняется добавление двух записей в таблицу "route". В столбец "number" вставляются значения 11 и 12.

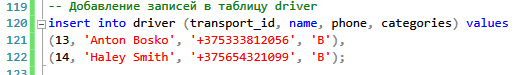
1. Добавление записей в таблицу transport



В этом запросе выполняется добавление двух записей в таблицу "transport". В столбцы "route\_id", "type", "gov\_num", "brand", "capacity" и "bus\_num" вставляются следующие значения:

* 11, 'Bus', '1234AB-7', 'МАЗ', 60, 1011
* 12, 'Tram', '5678CD-7', 'МАЗ', 10, 1012

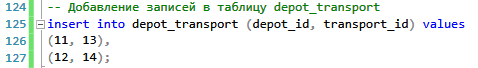
1. Добавление записей в таблицу driver



В этом запросе выполняется добавление двух записей в таблицу "driver". В столбцы "transport\_id", "name", "phone" и "categories" вставляются следующие значения:

* 13, 'Anton Bosko', '+375333812056', 'B'
* 14, 'Haley Smith', '+375654321099', 'B'

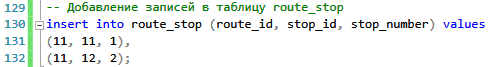
1. Добавление записей в таблицу depot\_transport



В этом запросе выполняется добавление двух записей в таблицу "depot\_transport". В столбцы "depot\_id" и "transport\_id" вставляются следующие значения:

* 11, 13
* 12, 14

1. Добавление записей в таблицу route\_stop

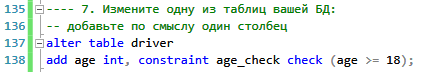


В этом запросе выполняется добавление двух записей в таблицу "route\_stop". В столбцы "route\_id", "stop\_id" и "stop\_number" вставляются следующие значения:

* 11, 11, 1
* 11, 12, 2

## Задание 7

Изменение одной из таблиц вашей БД добавлением нового столбца в таблицу driver



В этой команде выполняется изменение таблицы "driver" путем добавления нового столбца "age" с типом данных "int". Также указывается ограничение "age\_check", которое проверяет, что значения в столбце "age" больше или равны 18.

## Задание 8

Запросы, использующие агрегатные функции

1. Вычисление суммы значений столбца "age" из таблицы "driver"



В этом запросе выполняется вычисление суммы значений столбца "age" из таблицы "driver". Результат выводится в столбце "sum\_age".

1. Поиск максимального значения столбца "age" из таблицы "driver"



В этом запросе выполняется поиск максимального значения столбца "age" из таблицы "driver". Результат выводится в столбце "max\_age".

1. Поиск минимального значения столбца "age" из таблицы "driver"



В этом запросе выполняется поиск минимального значения столбца "age" из таблицы "driver". Результат выводится в столбце "min\_age".

1. Вычисление среднего значения столбца "age" из таблицы "driver"



В этом запросе выполняется вычисление среднего значения столбца "age" из таблицы "driver". Результат выводится в столбце "avg\_age".

5. Подсчетколичества уникальных значений столбца "type" в таблице "transport"

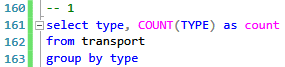


В этом запросе выполняется подсчет количества уникальных значений столбца "type" в таблице "transport". Результат выводится в столбце "types\_count".

## Задание 9

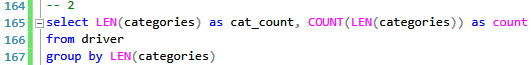
Запросы на группировку данных, используя оператор GROUP BY

1. Группировка данных в таблице "transport" по столбцу "type" и подсчет количества записей в каждой группе



В этом запросе выполняется группировка данных в таблице "transport" по значению столбца "type". Затем с помощью функции COUNT() подсчитывается количество записей в каждой группе. Результаты выводятся в столбцах "type" и "count", где "type" содержит уникальные значения столбца "type", а "count" содержит количество записей в каждой группе.

2. Группировка данных в таблице "driver" по длине значения столбца "categories" и подсчет количества записей в каждой группе

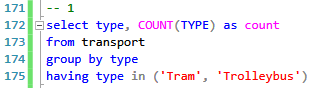


В этом запросе выполняется группировка данных в таблице "driver" по длине значений столбца "categories". С помощью функции LEN() определяется длина каждого значения столбца "categories". Затем с помощью функции COUNT() подсчитывается количество записей в каждой группе. Результаты выводятся в столбцах "cat\_count" и "count", где "cat\_count" содержит длину значений столбца "categories", а "count" содержит количество записей в каждой группе.

## Задание 10

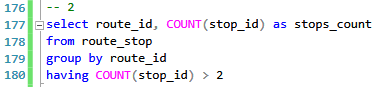
Запросы на фильтрацию групп с использованием оператора HAVING

1. Фильтрация групп в таблице "transport" по значению столбца "type"



В этом запросе выполняется группировка данных в таблице "transport" по значению столбца "type". Затем с помощью оператора HAVING происходит фильтрация групп, где значение столбца "type" равно 'Tram' или 'Trolleybus'. Только группы, удовлетворяющие указанному условию, будут включены в результаты.

1. Фильтрация групп в таблице "route\_stop" по количеству записей в каждой группе

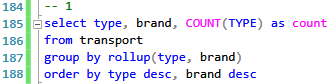


В этом запросе выполняется группировка данных в таблице "route\_stop" по значению столбца "route\_id". Затем с помощью оператора HAVING происходит фильтрация групп, где количество записей в столбце "stop\_id" больше 2. Только группы, удовлетворяющие указанному условию, будут включены в результаты.

## Задание 11

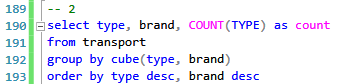
Запросы на расширения SQL Server для группировки (ROLLUP, CUBE, GROUPING SETS, OVER)

1. Расширение ROLLUP для группировки данных в таблице "transport"



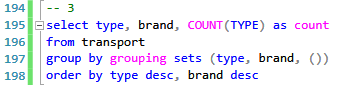
В этом запросе выполняется группировка данных в таблице "transport" с использованием расширения ROLLUP. Группировка происходит по столбцам "type" и "brand". Результаты выводятся в столбцах "type", "brand" и "count", где "count" содержит количество записей в каждой группе. Результаты будут упорядочены по убыванию значений столбцов "type" и "brand".

1. Расширение CUBE для группировки данных в таблице "transport"



В этом запросе выполняется группировка данных в таблице "transport" с использованием расширения CUBE. Группировка происходит по столбцам "type" и "brand". Результаты выводятся в столбцах "type", "brand" и "count", где "count" содержит количество записей в каждой группе. Результаты будут упорядочены по убыванию значений столбцов "type" и "brand".

1. Расширение GROUPING SETS для группировки данных в таблице "transport"



В этом запросевыполняется группировка данных в таблице "transport" с использованием расширения GROUPING SETS. Группировка происходит по столбцам "type" и "brand", а также по пустому множеству (пустой набор столбцов). Результаты выводятся в столбцах "type", "brand" и "count", где "count" содержит количество записей в каждой группе. Результаты будут упорядочены по убыванию значений столбцов "type" и "brand".

1. Использование функции WINDOW (OVER) для вычисления суммы возраста внутри каждой категории



В этом запросе выполняется выборка данных из таблицы "driver". Для каждой записи вычисляется сумма возраста (столбец "age") внутри каждой категории (столбец "categories") с использованием функции SUM() и оконной функции (WINDOW) OVER. Результаты выводятся в столбцах "name", "categories" и "sum\_of\_ages\_same\_cat", где "sum\_of\_ages\_same\_cat" содержит сумму возраста внутри каждой категории.

## Задание 12

Запросы на разворачивание данных (PIVOT) и отмену разворачивания данных (UNPIVOT)

PIVOT: Разворачивание данных в таблице "transport"



В этом запросе выполняется разворачивание данных в таблице "transport" с использованием оператора PIVOT. Сначала происходит выборка столбцов "brand" и "type" из таблицы "transport" в подзапросе. Затем оператор PIVOT используется для разворачивания значений столбца "type" в отдельные столбцы ("Bus", "Tram", "Trolleybus", "Electrobus"), а значения столбца "brand" используются в качестве идентификаторов строк. Результаты выводятся в столбцах "gov\_num", "name" и столбцах, соответствующих разворачиваемым значениям столбца "type".

UNPIVOT: Отмена разворачивания данных в таблице "transport"



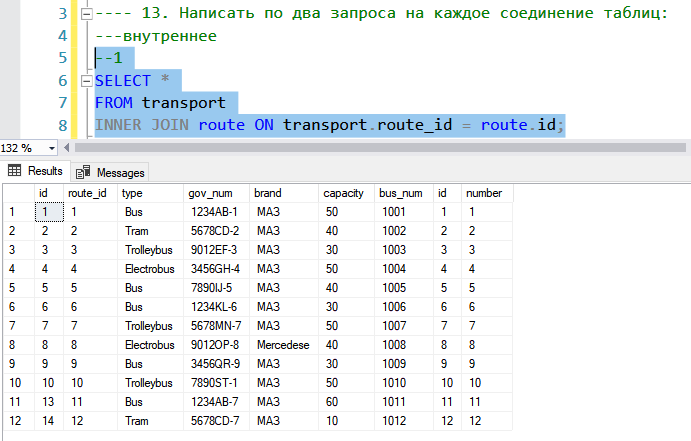
В этом запросе выполняется отмена разворачивания данных в таблице "transport" с использованием оператора UNPIVOT. Сначала происходит выборка столбцов "gov\_num", "brand" и "type" из таблицы "transport" в подзапросе. Затем оператор UNPIVOT используется для отмены разворачивания значений столбцов "brand" и "type" и преобразования их в отдельные строки с соответствующими значениями "name" и "value". Результаты выводятся в столбцах "gov\_num", "name" и "value".

# Лабораторная работа 5-6

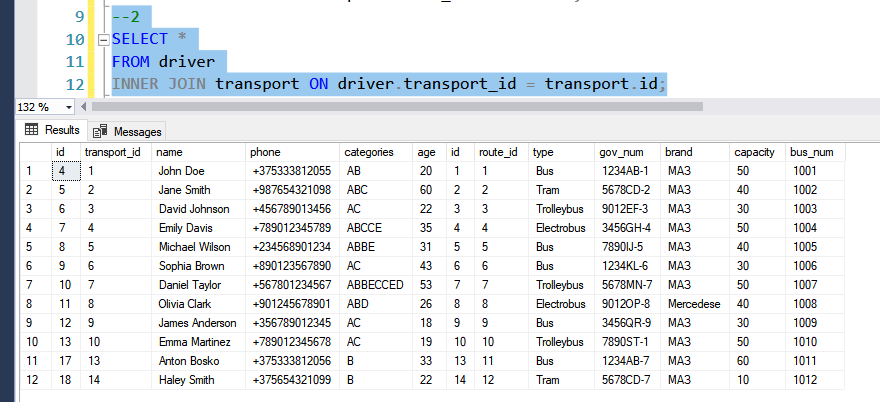
## Задание 13

Внутреннее соединение:

* Запрос 1: Выбрать все записи из таблицы "transport", которые имеют соответствующую запись в таблице "route" с помощью оператора INNER JOIN.

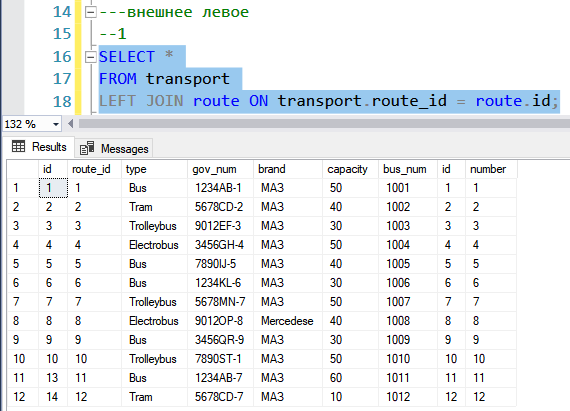


* Запрос 2: Выбрать все записи из таблицы "driver", которые имеют соответствующую запись в таблице "transport" с помощью оператора INNER JOIN.

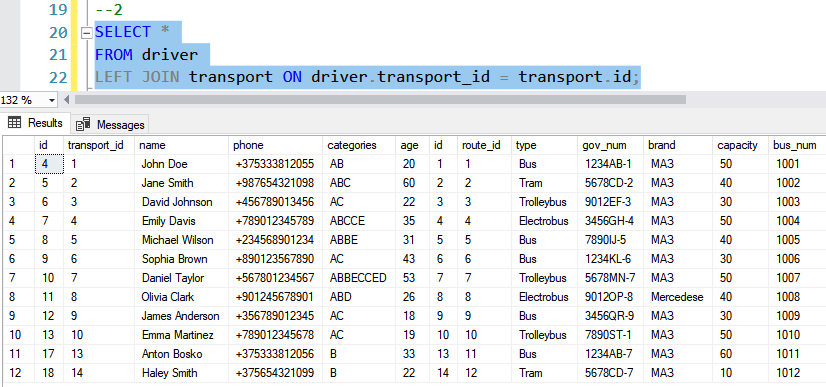


Внешнее левое соединение:

* Запрос 1: Выбрать все записи из таблицы "transport" и соответствующие записи из таблицы "route" с использованием оператора LEFT JOIN.

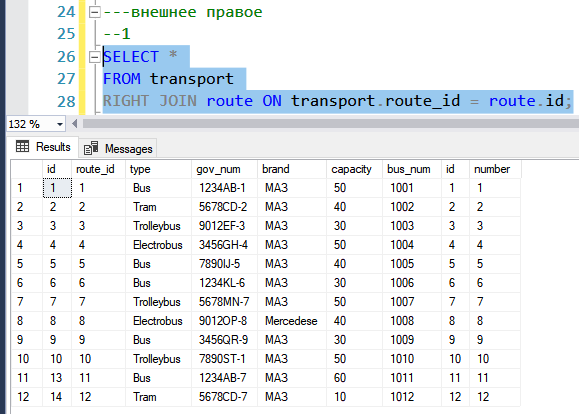


* Запрос 2: Выбрать все записи из таблицы "driver" и соответствующие записи из таблицы "transport" с использованием оператора LEFT JOIN.

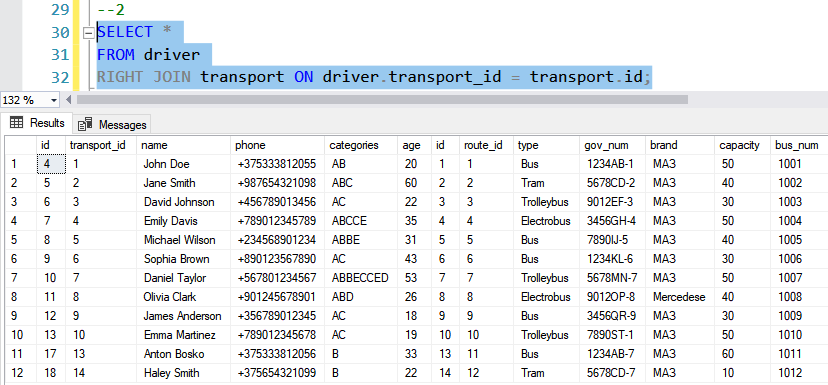


Внешнее правое соединение:

* Запрос 1: Выбрать все записи из таблицы "transport" и соответствующие записи из таблицы "route" с использованием оператора RIGHT JOIN.

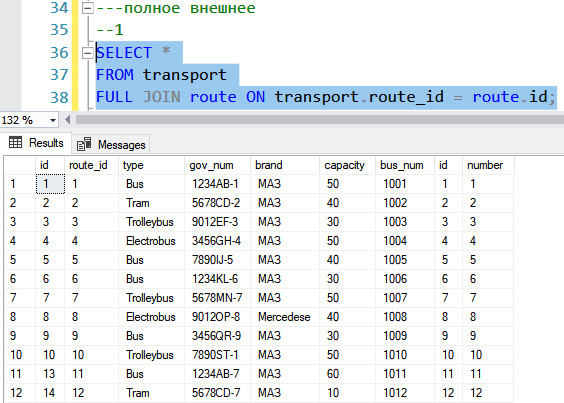


* Запрос 2: Выбрать все записи из таблицы "driver" и соответствующие записи из таблицы "transport" с использованием оператора RIGHT JOIN.

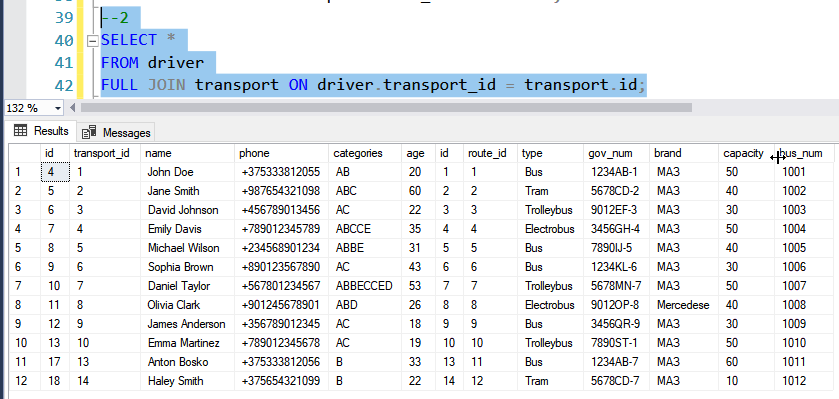


Полное внешнее соединение:

* Запрос 1: Выбрать все записи из таблицы "transport" и соответствующие записи из таблицы "route" с использованием оператора FULL JOIN.

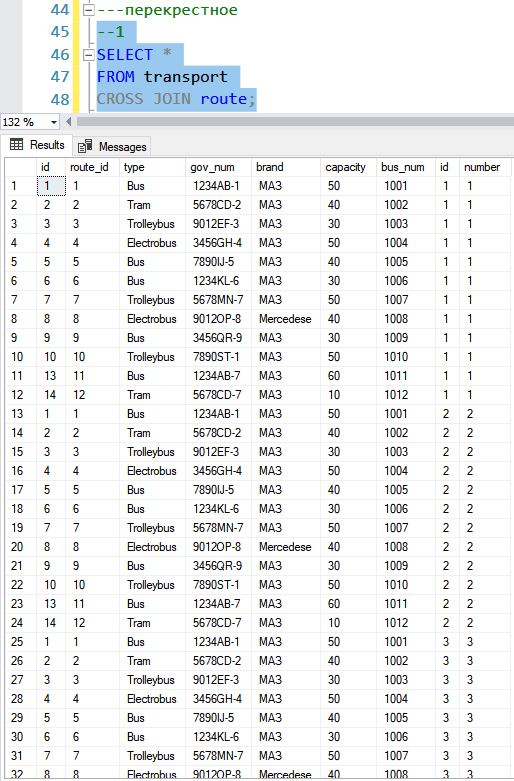


* Запрос 2: Выбрать все записи из таблицы "driver" и соответствующие записи из таблицы "transport" с использованием оператора FULL JOIN.

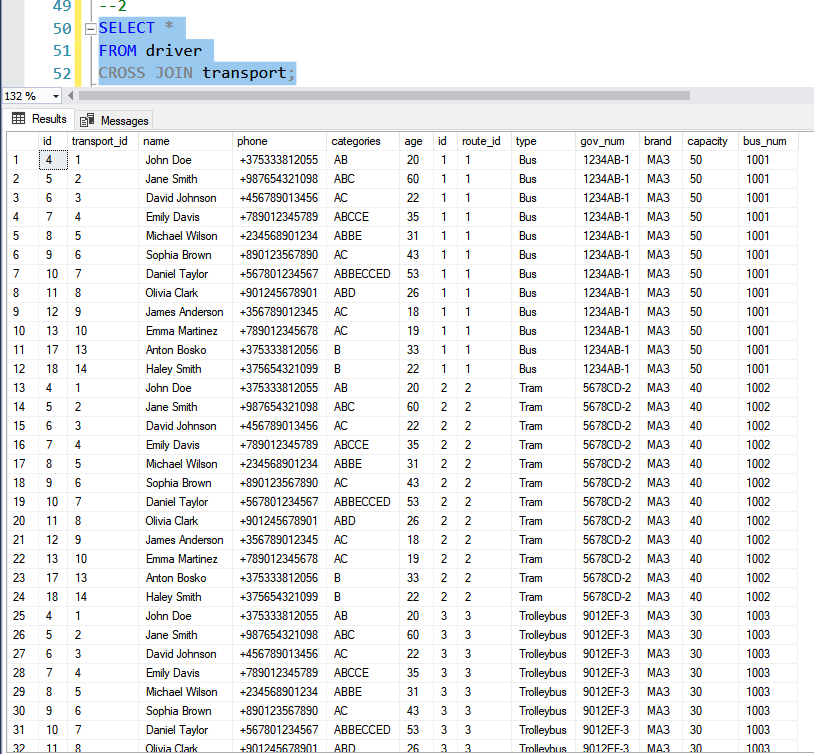


Перекрестное соединение:

* Запрос 1: Получить все возможные комбинации записей из таблиц "transport" и "route" с помощью оператора CROSS JOIN.



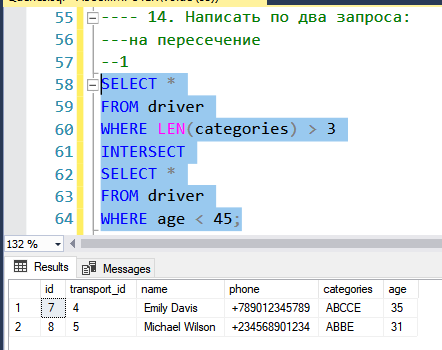
* Запрос 2: Получить все возможные комбинации записей из таблиц "driver" и "transport" с помощью оператора CROSS JOIN.



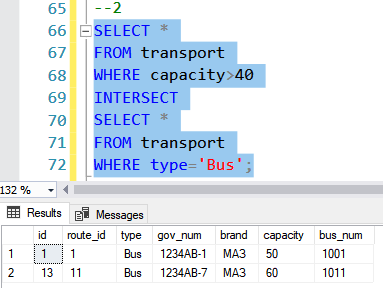
## Задание 14

Запросы на пересечение:

* Запрос 1: Выбрать все записи из таблицы "driver", у которых длина значения в столбце "categories" больше 3, и пересекающиеся с ними записи из таблицы "driver", у которых возраст меньше 45.

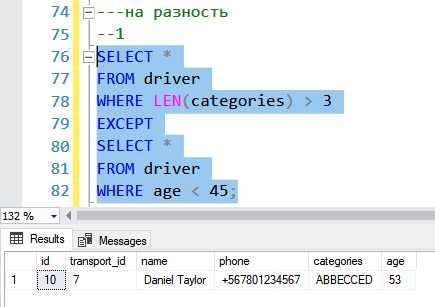


* Запрос 2: Выбрать все записи из таблицы "transport", у которых вместимость больше 40, и пересекающиеся с ними записи из таблицы "transport", у которых тип равен 'Bus'.

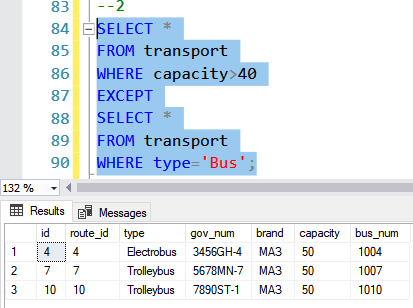


Запросы на разность:

* Запрос 1: Выбрать все записи из таблицы "driver", у которых длина значения в столбце "categories" больше 3, и не входящие в результаты запроса, выбирающего записи из таблицы "driver", у которых возраст меньше 45.

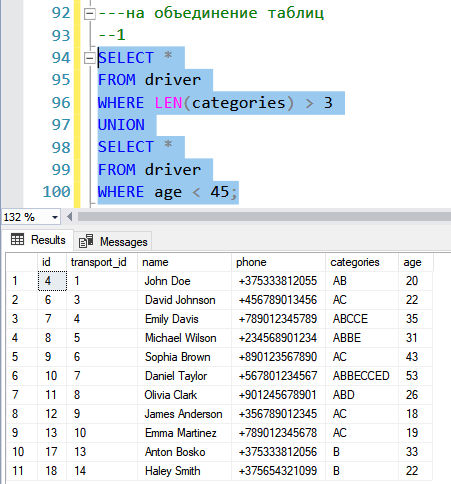


* Запрос 2: Выбрать все записи из таблицы "transport", у которых вместимость больше 40, и не входящие в результаты запроса, выбирающего записи из таблицы "transport", у которых тип равен 'Bus'.

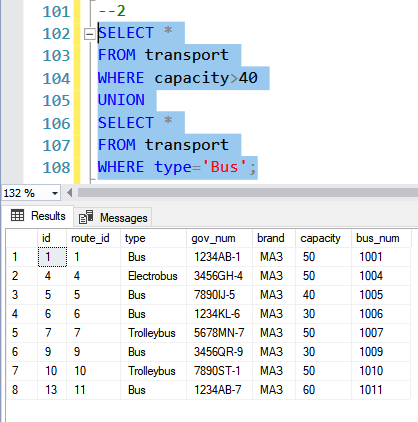


Запросы на объединение таблиц:

* Запрос 1: Выбрать все записи из таблицы "driver", у которых длина значения в столбце "categories" больше 3, и объединить с ними записи из таблицы "driver", у которых возраст меньше 45.

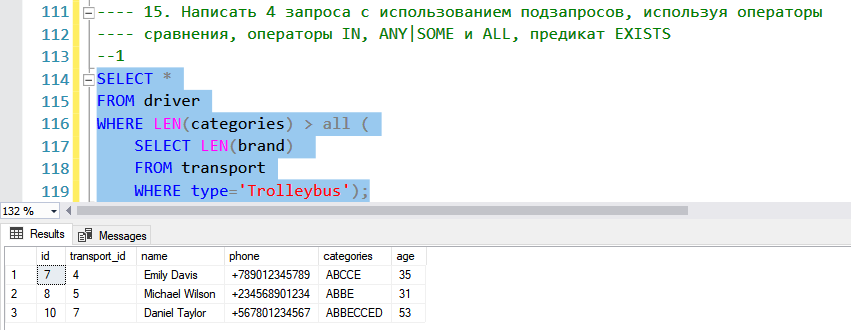


* Запрос 2: Выбрать все записи из таблицы "transport", у которых вместимость больше 40, и объединить с ними записи из таблицы "transport", у которых тип равен 'Bus'.

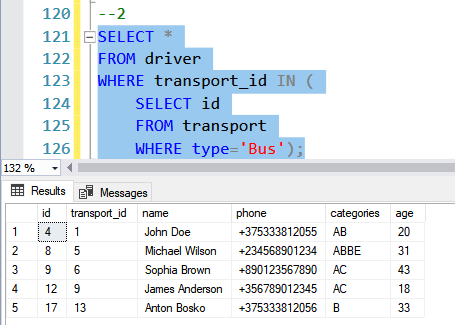


## Задание 15

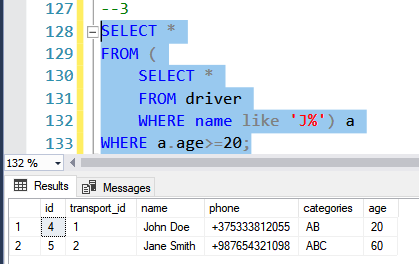
* Запрос 1: Выбрать все записи из таблицы "driver", у которых длина значения в столбце "categories" больше всех значений длиныв столбце "brand" в таблице "transport", где тип равен 'Trolleybus'.



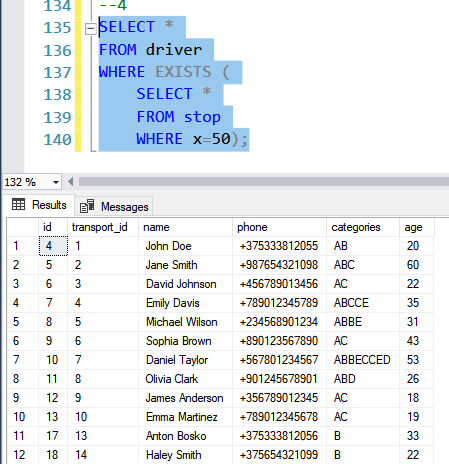
* Запрос 2: Выбрать все записи из таблицы "driver", у которых "transport\_id" содержится в результатах подзапроса, выбирающего значения столбца "id" из таблицы "transport", где тип равен 'Bus'.



* Запрос 3: Выбрать все записи из таблицы "driver", где имя начинается с буквы 'J', и из этих записей выбрать те, у которых возраст равен или больше 20.



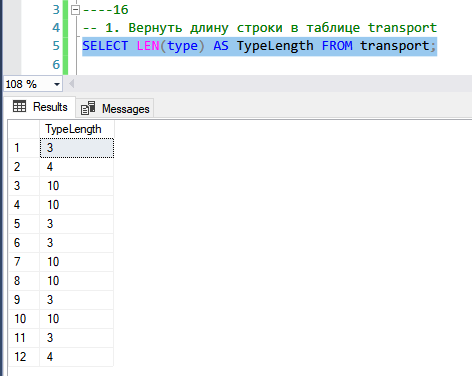
* Запрос 4: Выбрать все записи из таблицы "driver", у которых существуют записи в таблице "stop", где значение столбца "x" равно 50.



# Лабораторная работа 7-9

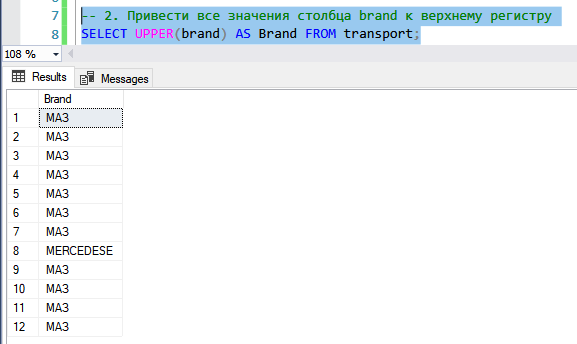
## Задание 16

1. Вернуть длину строки в таблице transport:



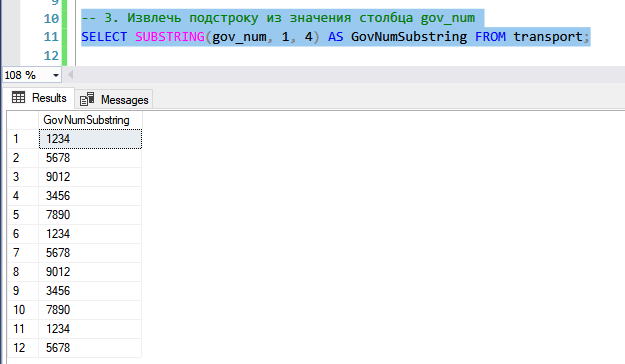
Результат: Таблица "transport" с дополнительным столбцом "TypeLength", содержащим длину каждого значения в столбце "type".

1. Привести все значения столбца brand к верхнему регистру:



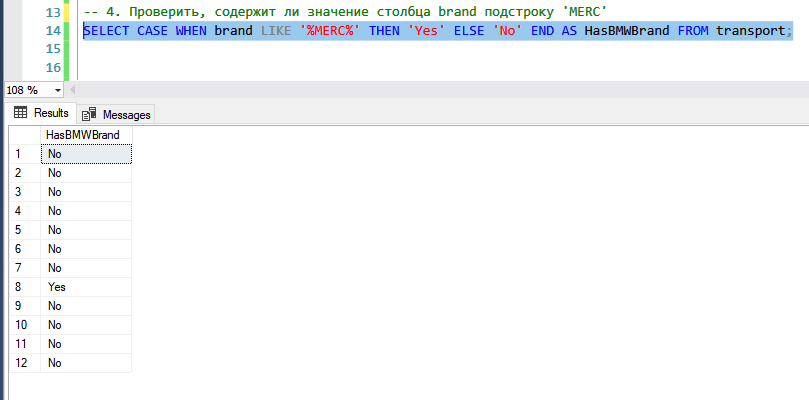
Результат: Таблица "transport" с дополнительным столбцом "Brand", содержащим значения столбца "brand", приведенные к верхнему регистру.

1. Извлечь подстроку из значения столбца gov\_num:



Результат: Таблица "transport" с дополнительным столбцом "GovNumSubstring", содержащим первые четыре символа из значения столбца "gov\_num".

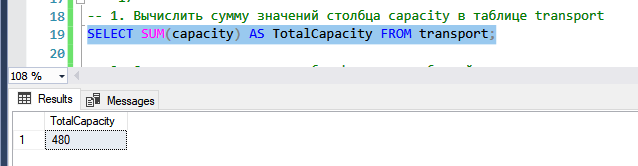
1. Проверить, содержит ли значение столбца brand подстроку MERC:



Результат: Таблица "transport" с дополнительным столбцом "HasBMWBrand", содержащим значения 'Yes' для тех строк, в которых значение столбца "brand" содержит подстроку MERC, и значения 'No' для остальных строк.

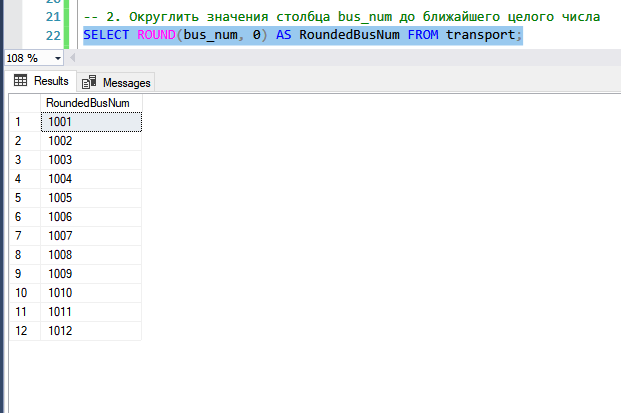
## Задание 17

1. Вычислить сумму значений столбца capacity в таблице transport:



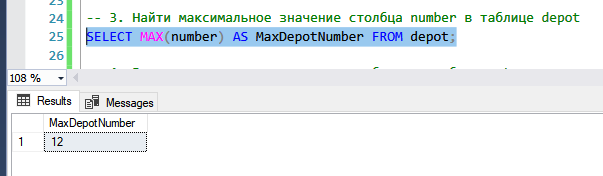
Результат: Одна строка с суммой значений столбца "capacity" из таблицы "transport".

1. Округлить значения столбца bus\_num до ближайшего целого числа:



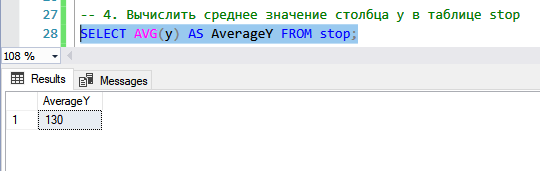
Результат: Таблица "transport" с дополнительным столбцом "RoundedBusNum", содержащим значения столбца "bus\_num", округленные до ближайшего целого числа.

1. Найти максимальное значение столбца number в таблице depot:



Результат: Одна строка с максимальным значением столбца "number" из таблицы "depot".

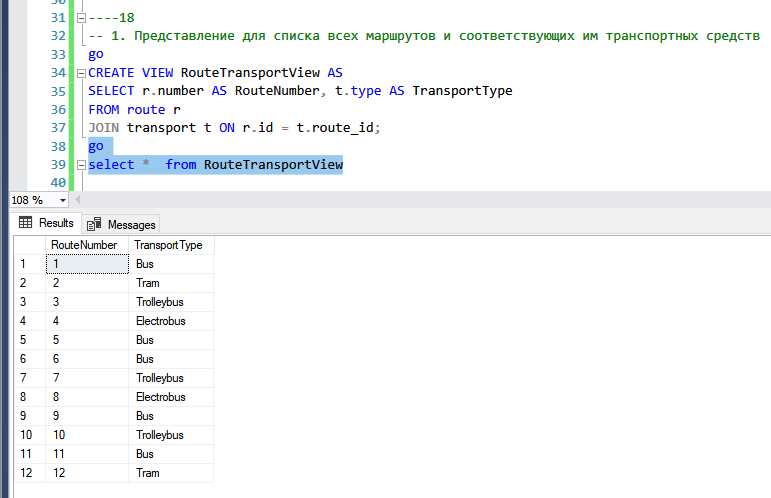
1. Вычислить среднее значение столбца y в таблице stop:



Результат: Одна строка со средним значением столбца "y" из таблицы "stop".

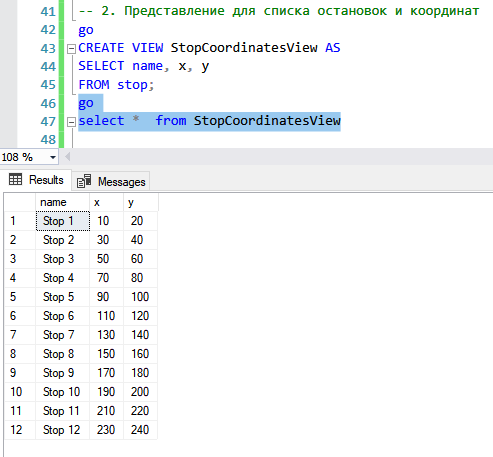
## Задание 18

1. Представление для списка всех маршрутов и соответствующих им транспортных средств:



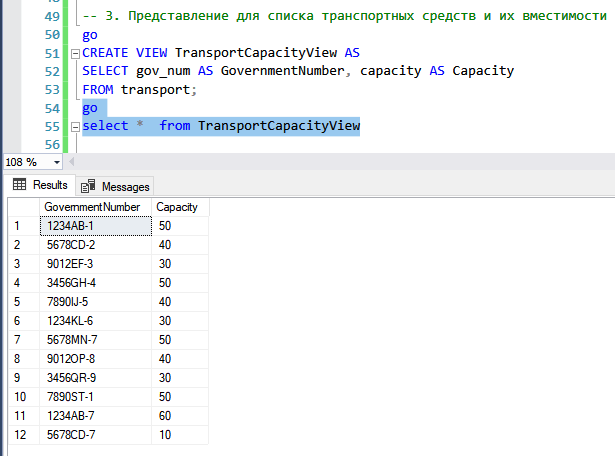
Результат: Представление "RouteTransportView" содержит два столбца: "RouteNumber" с номерами маршрутов и "TransportType" с типами транспортных средств, соответствующих каждому маршруту.

1. Представление для списка остановок и координат:



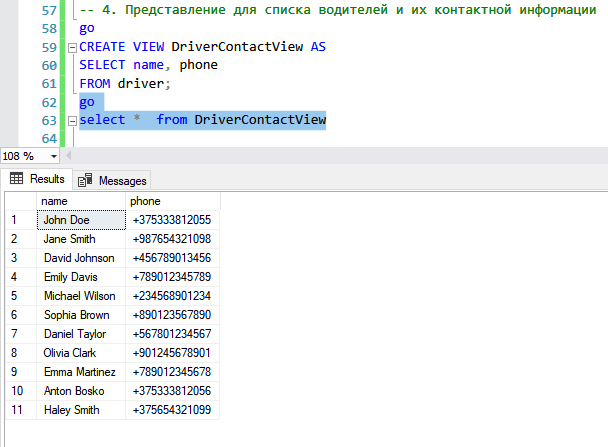
Результат: Представление "StopCoordinatesView" содержит три столбца: "name" с названиями остановок, "x" с координатами X и "y" с координатами Y каждой остановки.

1. Представление для списка транспортных средств и их вместимости:



Результат: Представление "TransportCapacityView" содержит два столбца: "GovernmentNumber" с государственными номерами транспортных средств и "Capacity" с их вместимостью.

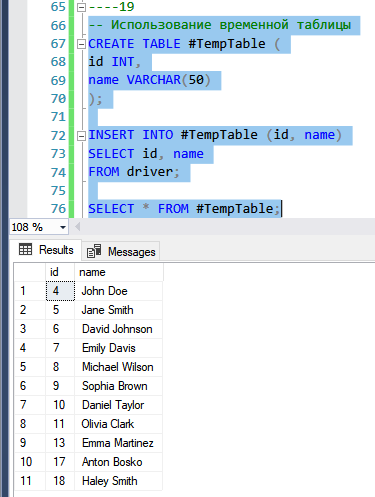
1. Представление для списка водителей и их контактной информации:



Результат: Представление "DriverContactView" содержит два столбца: "name" с именами водителей и "phone" с их контактными номерами телефонов.

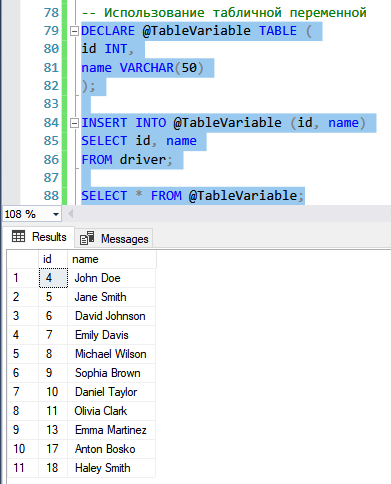
## Задание 19

Использование временной таблицы:



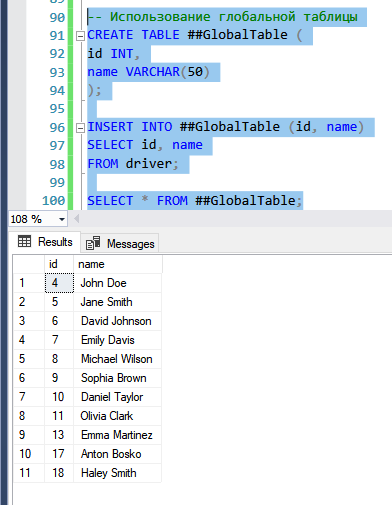
Результат: Создается временная таблица "#TempTable" с двумя столбцами "id" и "name". Затем данные из таблицы "driver" вставляются во временную таблицу, и результат выводится.

Использование табличной переменной:



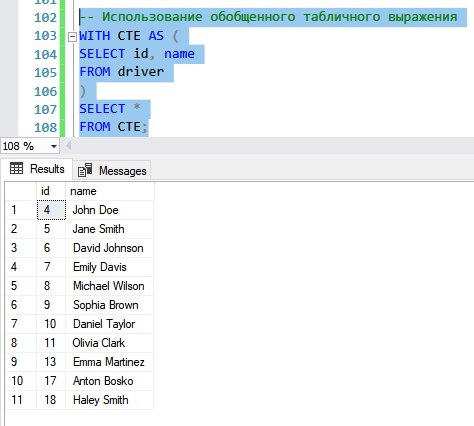
Результат: Объявляется табличная переменная "@TableVariable" с двумя столбцами "id" и "name". Затем данные из таблицы "driver" вставляются в табличную переменную, и результат выводится.

Использование глобальной таблицы:



Результат: Создается глобальная таблица "##GlobalTable" с двумя столбцами "id" и "name". Затем данные из таблицы "driver" вставляются в глобальную таблицу, и результат выводится.

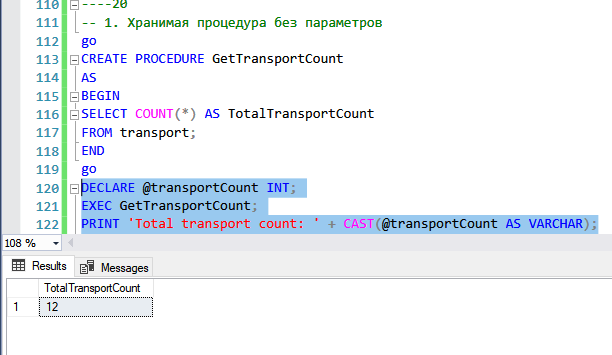
Использование обобщенного табличного выражения:



Результат: Объявляется обобщенное табличное выражение "CTE" с двумя столбцами "id" и "name", которое представляет данные из таблицы "driver". Затем результат выводится.

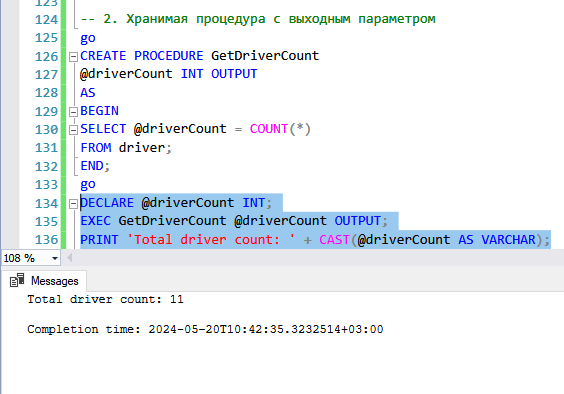
## Задание 20

1. Хранимая процедура без параметров:



Результат: Создается хранимая процедура "GetTransportCount" без параметров, которая возвращает количество записей в таблице "transport". Затем выполняется процедура, и результат выводится.

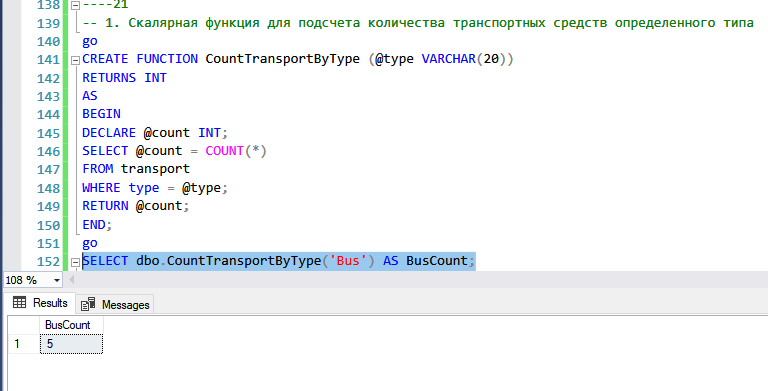
1. Хранимая процедура с выходным параметром:



Результат: Создается хранимая процедура "GetDriverCount" с выходным параметром "@driverCount", которая возвращает количество записей в таблице "driver". Затем выполняется процедура, и результат выводится.

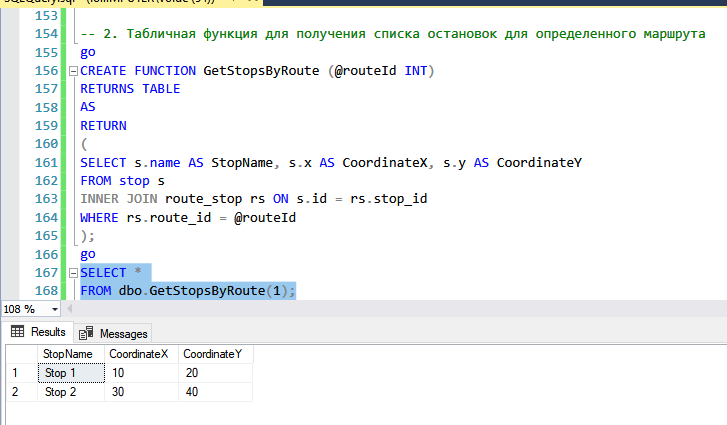
## Задание 21

1. Скалярная функция для подсчета количества транспортных средств определенного типа:



Результат: Создается скалярная функция "CountTransportByType", которая принимает тип транспортного средства в качестве параметра и возвращает количество транспортных средств с указанным типом. Затем функция вызывается, и результат выводится.

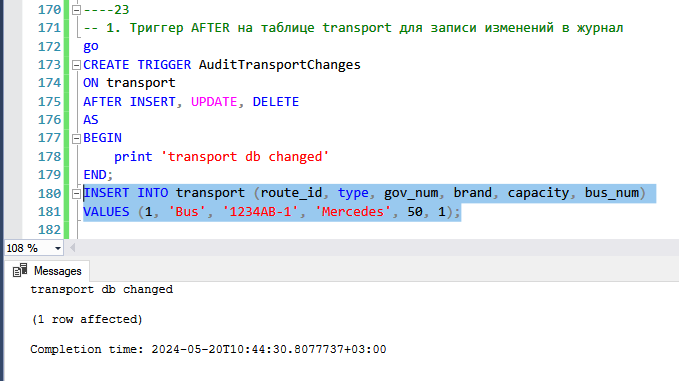
1. Табличная функция для получения списка остановок для определенного маршрута:



Результат: Создается табличная функция "GetStopsByRoute", которая принимает идентификатор маршрута в качестве параметра и возвращает список остановок для указанного маршрута. Затем функция вызывается, и результат выводится.

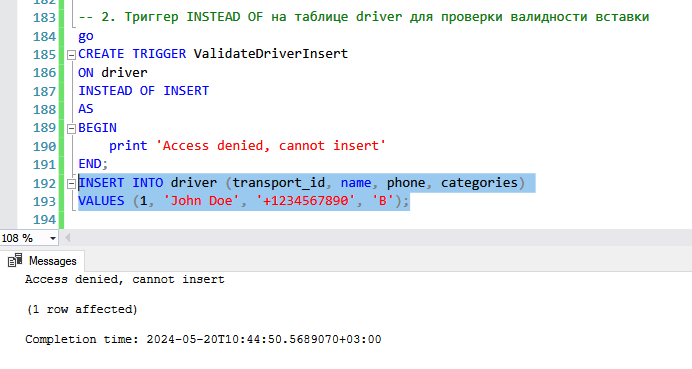
## Задание 23

1. Триггер AFTER на таблице transport для записи изменений в журнал:



Результат: Создается триггер "AuditTransportChanges" на таблице "transport", который срабатывает после операций вставки, обновления и удаления и выводит сообщение "transport db changed" при каждом срабатывании.

1. Триггер INSTEAD OF на таблице driver для проверки валидности вставки:



Результат: Создается триггер "ValidateDriverInsert" на таблице "driver", который срабатывает вместо операции вставки и выводит сообщение "Access denied, cannot insert" при попытке вставки в таблицу "driver".