БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

**ОТЧЕТ**

по индивидуальному заданию

учебной дисциплины

«Модели данных и системы управления базами данных»

Дополнение к лабораторным работам №2-4

**Выполнила:**

Горбач Владимир Игоревич

**Преподаватель:**

Дрепакова Ангелина Валерьевна

Минск, 2024

**Содержание**

[**Задание 1** 3](#_Toc162297985)

[**Задание 2** 4](#_Toc162297986)

[**Задание 3** 5](#_Toc162297987)

[**Задание 4** 8](#_Toc162297988)

[**Задание 5** 10](#_Toc162297989)

[**Задание 6** 11](#_Toc162297990)

[**Задание 7** 13](#_Toc162297991)

[**Задание 8** 14](#_Toc162297992)

[**Задание 9** 15](#_Toc162297993)

[**Задание 10** 16](#_Toc162297994)

[**Задание 11** 17](#_Toc162297995)

[**Задание 12** 19](#_Toc162297996)

# **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.

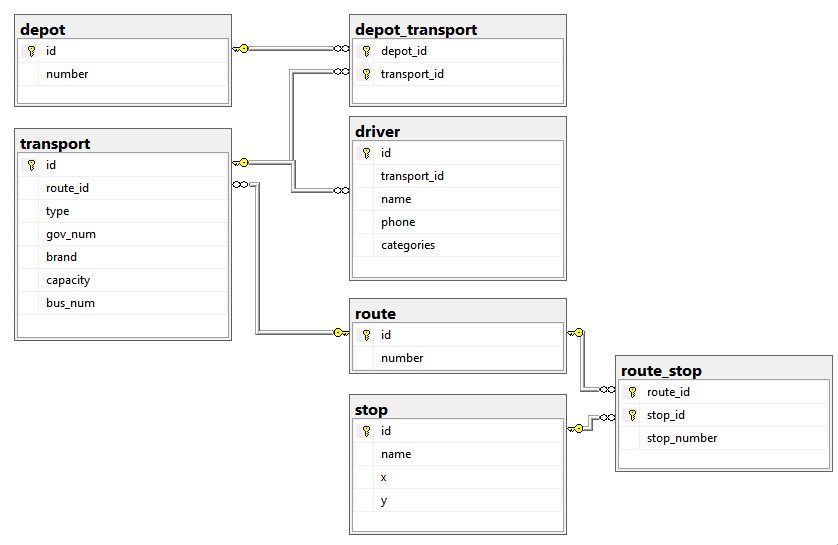
• Остановки:

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.

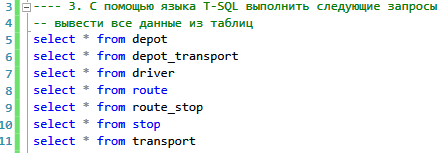
# **Задание 2**



# **Задание 3**

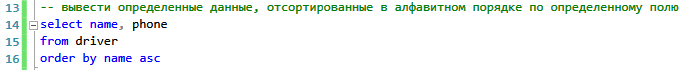
Очень рад, что получил еще одну попытку помочь вам! Вот подробные пояснения для каждого запроса на языке T-SQL:

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

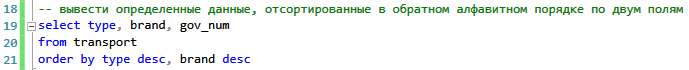


Этот запрос выбирает все строки и все столбцы из таблицы "depot", "depot\_transport", "driver", "route", "route\_stop", "stop" и "transport" и выводит их на экран.

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":



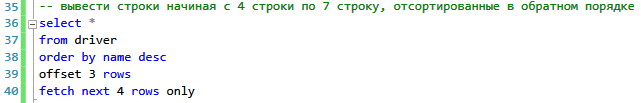
Этот запрос выбирает верхние 30% строк из таблицы "stop" и выводит все столбцы этих строк на экран.

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

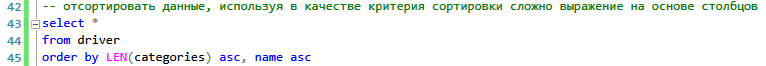


Этот запрос выбирает верхние 5 строк из таблицы "driver" и выводит все столбцы этих строк на экран.

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

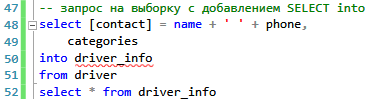
Этот запрос выбирает строки с 4 по 7 из таблицы "driver". Результаты сортируются в обратном порядке по столбцу "name" и пропускаются первые 3 строки с помощью OFFSET, затем выбираются следующие 4 строки с помощью FETCH NEXT.

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".

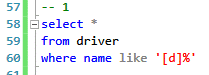
9. Запрос на выборку с добавлением SELECT INTO



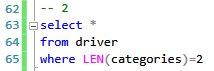
В этом запросе выбираются столбцы "name" и "phone" из таблицы "driver". Каждая выбранная запись объединяется в одно значение в столбце "contact", где имя и номер телефона разделены пробелом. Затем выбранные столбцы "contact" и "categories" вставляются в новую таблицу "driver\_info" с помощью оператора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

# **Задание 4**

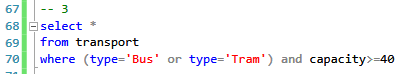
1. В этом запросе выбираются все столбцы из таблицы "driver", где значение в столбце "name" начинается с буквы "d" (используется шаблон "[d]%").



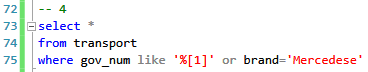
2. В этом запросе выбираются все столбцы из таблицы "driver", где длина значения в столбце "categories" равна 2.



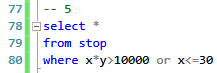
3. В этом запросе выбираются все столбцы из таблицы "transport", где значение в столбце "type" равно "Bus" или "Tram", а значение в столбце "capacity" больше или равно 40.



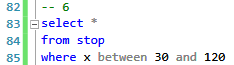
4. В этом запросе выбираются все столбцы из таблицы "transport", где значение в столбце "gov\_num" оканчивается на цифру "1" или значение в столбце "brand" равно "Mercedes".



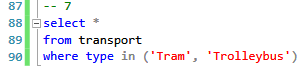
5. В этом запросе выбираются все столбцы из таблицы "stop", где произведение значений в столбцах "x" и "y" больше 10000 или значение в столбце "x" меньше или равно 30.



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

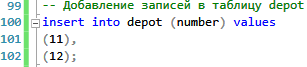


В этой команде выполняется переименование таблицы "driver\_info" в "driver\_information" с помощью хранимой процедуры sp\_rename. Первый параметр указывает текущее имя таблицы, а второй параметр указывает новое имя таблицы.

# **Задание 6**

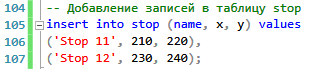
Добавление записей в каждую из таблиц вашей БД

1. Добавление записей в таблицу depot



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

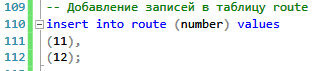
1. Добавление записей в таблицу stop



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

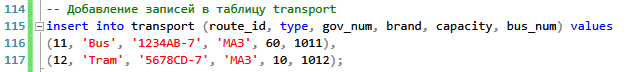
* 'Stop 11', 210, 220
* 'Stop 12', 230, 240

1. Добавление записей в таблицу route



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

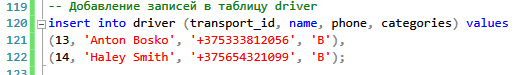
1. Добавление записей в таблицу transport



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

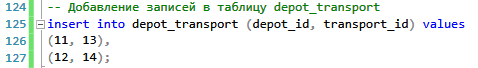
1. Добавление записей в таблицу driver



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'

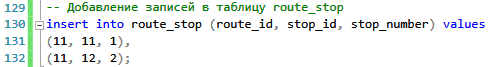
1. Добавление записей в таблицу depot\_transport



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

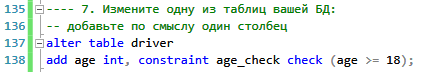


В этом запросе выполняется добавление двух записей в таблицу "route\_stop". В столбцы "route\_id", "stop\_id" и "stop\_number" вставляются следующие значения:

* 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"

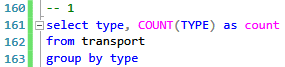


В этом запросе выполняется подсчет количества уникальных значений столбца "type" в таблице "transport". Результат выводится в столбце "types\_count".

# **Задание 9**

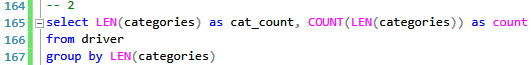
Запросы на группировку данных, используя оператор GROUP BY

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

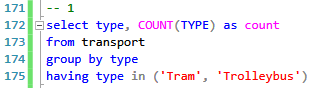


В этом запросе выполняется группировка данных в таблице "driver" по длине значений столбца "categories". С помощью функции LEN() определяется длина каждого значения столбца "categories". Затем с помощью функции COUNT() подсчитывается количество записей в каждой группе. Результаты выводятся в столбцах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

# **Задание 10**

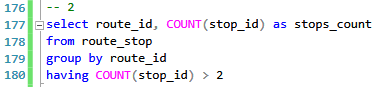
Запросы на фильтрацию групп с использованием оператора HAVING

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

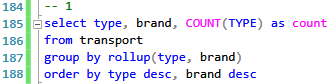


В этом запросе выполняется группировка данных в таблице "route\_stop" по значению столбца "route\_id". Затем с помощью оператора HAVING происходит фильтрация групп, где количество записей в столбце "stop\_id" больше 2. Только группы, удовлетворяющие указанному условию, будут включены в результаты.

# **Задание 11**

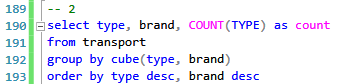
Запросы на расширения SQL Server для группировки (ROLLUP, CUBE, GROUPING SETS, OVER)

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



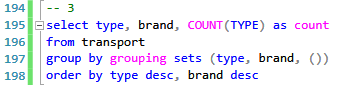
В этом запросе выполняется группировка данных в таблице "transport" с использованием расширения ROLLUP. Группировка происходит по столбцам "type" и "brand". Результаты выводятся в столбцах "type", "brand" и "count", где "count" содержит количество записей в каждой группе. Результаты будут упорядочены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