**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра ВТ**

отчет

**по индивидуальному домашнему заданию**

**по дисциплине «Базы данных»**

**Тема: Проектирование и создание базы данных «Управление торговли»**

|  |  |  |
| --- | --- | --- |
| Студента гр. 9308 |  | Яловега Н.В. |
| Преподаватель |  | Жирнова О.А. |

Санкт-Петербург

2021

**Аннотация**

В процессе выполнения индивидуального домашнего задания была спроектирована и создана база данных «Управление торговли». При проектировании и создании были использованы знания, полученные в процессе изучения курса «Базы данных». Были созданы и использованы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), построена диаграмма базы данных и выбрана стратегия резервного копирования.

**Summary**

In the process of completing the individual homework, the Trade Management database was designed. When designing, the knowledge gained in the course of studying the course "Databases" was used. Middleware objects (views, stored procedures, UDFs) were created and used, a database diagram was built, and a backup strategy was selected.

**содержание**

[ВВЕДЕНИЕ 4](#__RefHeading___Toc3540_4210544620)

[Цель работы 5](#__RefHeading___Toc3542_4210544620)

[Формулировка задания 5](#__RefHeading___Toc3544_4210544620)

[Формализация задачи 5](#__RefHeading___Toc3546_4210544620)

[Создание БД 6](#__RefHeading___Toc3548_4210544620)

[Создание таблиц и ограничений целостности 6](#__RefHeading___Toc3550_4210544620)

[Диаграмма 9](#__RefHeading___Toc3552_4210544620)

[Спецификация базы данных 10](#__RefHeading___Toc3554_4210544620)

[Разработка объектов промежуточного слоя 14](#__RefHeading___Toc3556_4210544620)

[Представления 14](#__RefHeading___Toc3558_4210544620)

[Хранимые процедуры 15](#__RefHeading___Toc3560_4210544620)

[UDF 17](#__RefHeading___Toc3562_4210544620)

[Резервное копирование 19](#__RefHeading___Toc3564_4210544620)

[Диаграмма 20](#__RefHeading___Toc3566_4210544620)

[Выводы 21](#__RefHeading___Toc3568_4210544620)

# ВВЕДЕНИЕ

В данной работе была спроектирована и создана база данных (БД) «Управление торговли», предназначенная для хранения и управления сведений о магазинах, отделах, товарах, торговых базах.

Также в процессе выполнения работы были созданы объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы), а также выбрана стратегия резервного копирования.

# Цель работы

Закрепить теоретические знания, полученные на курсе «Базы данных», и получить практические навыки в проектировании и создании базы данных. Необходимо создать объекты промежуточного слоя и определить стратегию резервного копирования.

# Формулировка задания

Для работы была выбрана база данных «Управление торговли». Необходимо выполнить следующие задания:

1. Спроектировать БД (структура данных)
2. Создать БД
3. Создать таблицы и ограничения целостности
4. Заполнить таблицы данными
5. Создать объекты промежуточного слоя (представления, хранимые процедуры, UDF-ы)
6. Создать стратегию резервного копирования

# Формализация задачи

Необходимо спроектировать базу данных, содержащую следующие таблицы:

1. bases – таблица, хранящая в себе информацию о торговых базах.
2. shops – таблица, хранящая в себе информацию о магазине.
3. departments – таблица, хранящая в себе информацию об отделах магазина.
4. products\_b – таблица, содержащая информацию о товарах, хранящихся на базе.
5. products\_d – таблица, содержащая информацию о товарах, хранящихся в отделе.
6. product\_department – таблица,
7. product\_base – таблица,

# Создание БД

Создание базы данных «Управление торговли»:

CREATE DATABASE [Trade]

CONTAINMENT = NONE

ON PRIMARY

( NAME = N'Trade', FILENAME = N'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Trade.mdf' , SIZE = 8192KB , MAXSIZE = UNLIMITED, FILEGROWTH = 65536KB )

LOG ON

( NAME = N'Trade\_log', FILENAME = N'D:\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\DATA\Trade\_log.ldf' , SIZE = 8192KB , MAXSIZE = 2048GB , FILEGROWTH = 65536KB )

WITH CATALOG\_COLLATION = DATABASE\_DEFAULT

GO

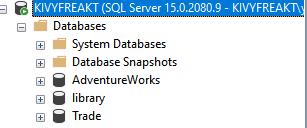


Рисунок 1 - Созданная база данных

# Создание таблиц и ограничений целостности

Создание таблицы bases:

Создание таблицы shops:

Создание таблицы departments:

Создание таблицы products\_b:

Создание таблицы products\_d:

Создание таблицы product\_department:

Создание таблицы product\_base:

На рис. 2 представлены созданные таблицы в базе данных Trade.

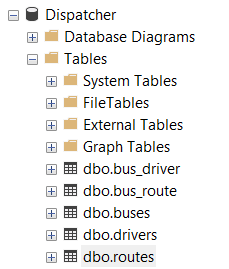
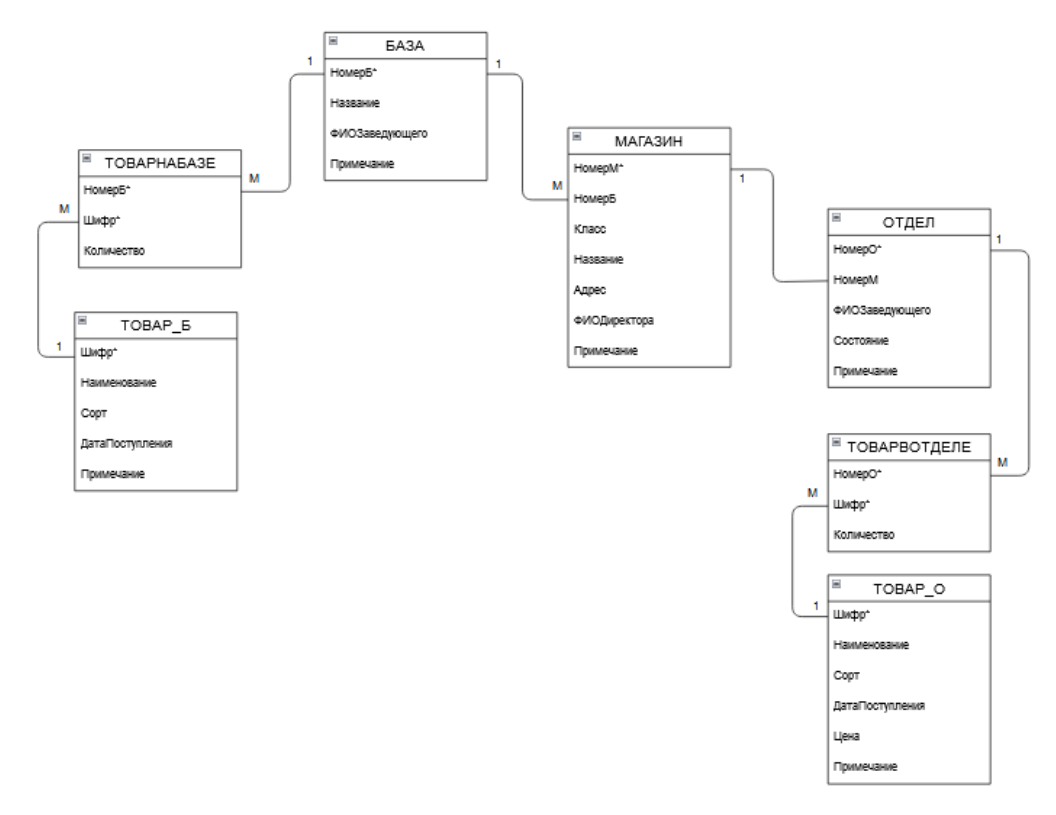


Рисунок 2 - Созданные таблицы

# Диаграмма

На рис. 3 представлена диаграмма связей между таблицами.

Рисунок 3 - Связи между таблицами

# Спецификация базы данных

Таблица 1 – Спецификация таблицы drivers

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица водителей** | | | | Имя таблицы:  drivers | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер таблицы:  **1** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Табельный номер | | number | int | | P |  | |  |
|  | 2 | Фамилия | | surname | nchar(20) | | notNull |  | |  |
|  | 3 | Имя | | name | nchar(20) | | notNull |  | |  |
|  | 4 | Отчество | | patronymic | nchar(20) | |  |  | |  |
|  | 5 | Опыт работы | | exp | smallint | | notNull | [exp]>(0) | |  |
|  | 6 | Примечание | | note | nchar(20) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 2 – Спецификация таблицы buses

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица автобусов** | | | | Имя таблицы:  buses | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер таблицы:  **2** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Гос. номер | | reg\_number | int | | P |  | |  |
|  | 2 | Тип | | type | nchar(15) | | notNull |  | |  |
|  | 3 | Вместительность | | capacity | int | | notNull | [capactiy]>(0) | |  |
|  | 4 | Состояние | | state | nchar(15) | | notNull |  | |  |
|  | 5 | Примечание | | note | nchar(15) | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 3 – Спецификация таблицы routes

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица маршрутов** | | | | Имя таблицы:  routes | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер таблицы:  **3** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Номер маршрута | | number | int | | P |  | |  |
|  | 2 | Время начала движения | | start\_time | time(0) | | notNull |  | |  |
|  | 3 | Время конца движения | | finish\_time | time(0) | | notNull |  | |  |
|  | 4 | Интервал движения | | interval | int | | notNull | [interval]>(0) | |  |
|  | 5 | Протяженность пути | | length | float | | notNull | [length]>(0) | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 4 – Спецификация таблицы bus\_driver

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица учёта наёма водителей** | | | | Имя таблицы:  bus\_driver | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер таблицы:  **4** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Гос. номер автобуса | | reg\_number | int | | P |  | |  |
|  | 2 | Табельный номер нанимаемого водителя | | driver\_number | int | | notNull |  | |  |
|  | 3 | Время наёма | | start\_time | datetime | | P, notNull |  | |  |
|  | 4 | Время увольнения | | finish\_time | datetime | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Таблица 5 – Спецификация таблицы bus\_route

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Описание структуры таблицы БД** | | | Наименование таблицы БД:  **Таблица учёта автобусов на маршрутах** | | | | Имя таблицы:  bus\_route | | | |
| Дата разработки:  17.11.2021 | | |
| Порядковый номер  таблицы:  **5** | | |  | | | |
|  |  |  | |  | |  |  | |  |  |
|  | № п/п | Наименование поля | | Спецификация данных | | | | | |  |
|  | Имя поля | Тип данных | | Ключ | Ограничения целостности | |  |
|  | 1 | Гос. номер автобуса | | reg\_number | int | | P, FK |  | |  |
|  | 2 | Табельный номер нанимаемого водителя | | driver\_number | int | | notNull, FK |  | |  |
|  | 3 | Время постановки на маршрут | | start\_time | datetime | | P, notNull |  | |  |
|  | 4 | Время снятия с маршрута | | finish\_time | datetime | |  |  | |  |
|  |  |  | |  |  | |  |  | |  |

Заполнение таблиц данными

Заполнение таблицы buses:

INSERT INTO buses

VALUES

(189023, 'Микроавтобус', '48', 'На работе', 'Исправен'),

(178023, 'Большой', '96', 'На работе', 'Исправен'),

(183623, 'Микроавтобус', '48', 'На стоянке', 'Исправен'),

(185023, 'Большой', '48', 'На стоянке', 'Исправен'),

(123423, 'Микроавтобус', '48', 'На стоянке', 'В ремонте');

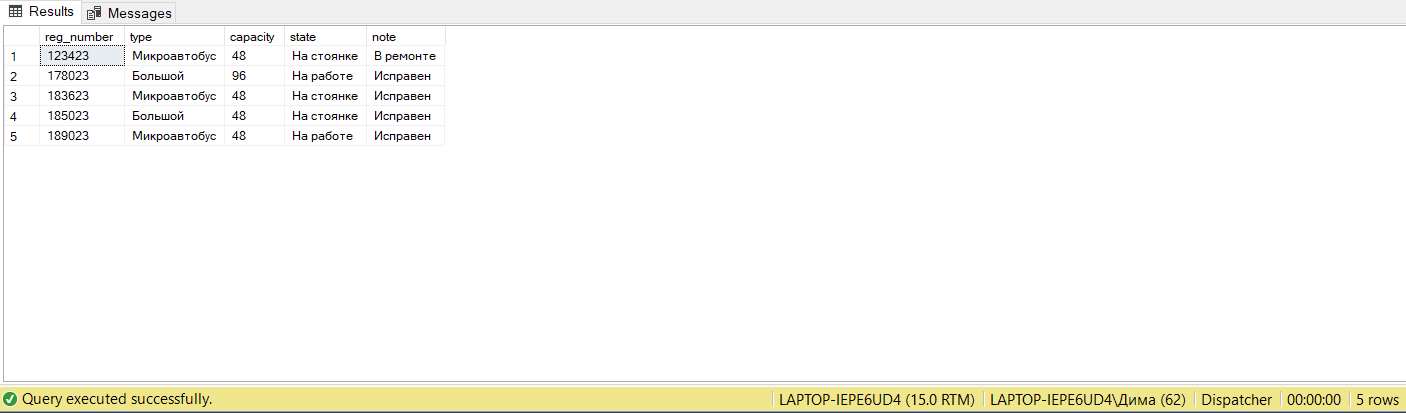


Рисунок 4 – Результат заполнения таблицы buses

Заполнение таблицы routes:

INSERT INTO routes

VALUES

(5, '09:00:00', '20:00:00', 14, 24),

(13, '10:00:00', '21:45:00', 10, 16),

(7, '08:00:00', '19:15:00', 9, 20),

(10, '08:05:00', '18:00:00', 14, 24),

(1, '13:45:00', '21:00:00', 25, 36);

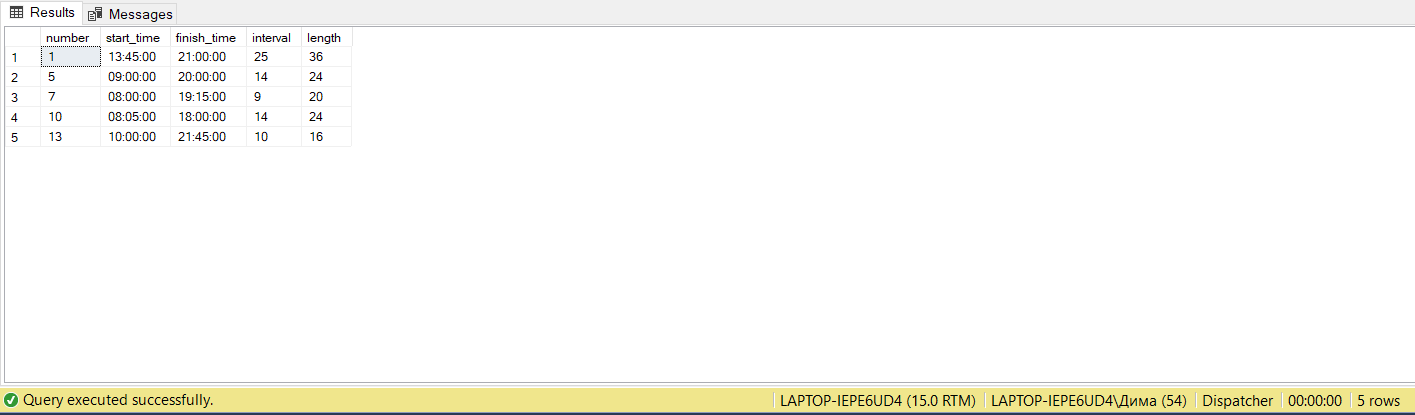


Рисунок 51 – Результат заполнения таблицы routes

Заполнение таблицы drivers:

INSERT INTO drivers

VALUES

(185, 'Дементьев', 'Дмитрий', 'Павлович', 1, 'В работе'),

(2673, 'Иванов', 'Сергей', 'Сергеевич', 5, 'В работе'),

(2564, 'Некотаев', 'Дмитрий', 'Рорев', 2, 'В работе'),

(198, 'Максимов', 'Никита', 'Рюрикович', 3, 'На больничном'),

(5923, 'Васнецов', 'Виктор', 'Викторович', 10, 'В работе');

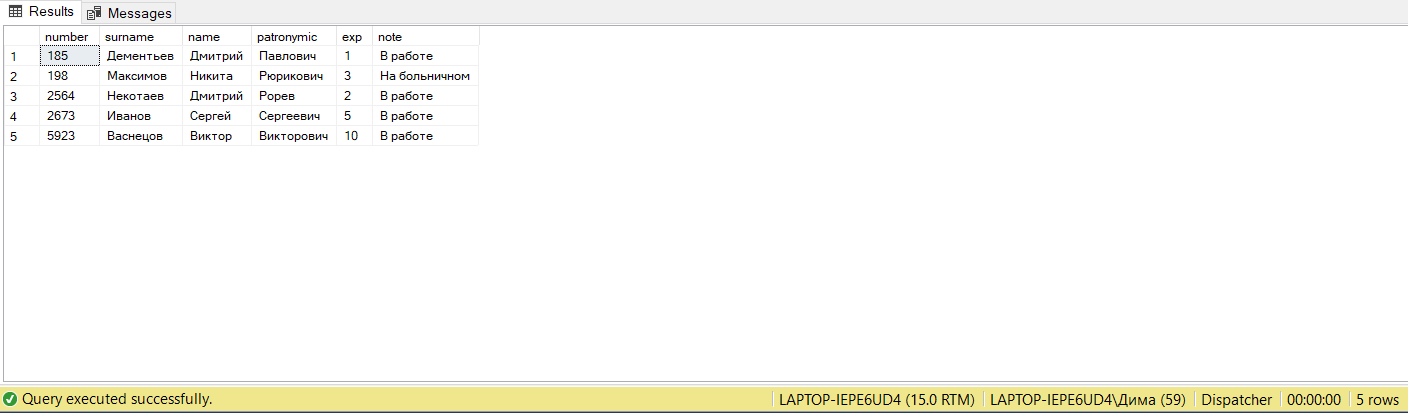


Рисунок 6 2– Результат заполнения таблицы bus\_driver

Заполнение таблицы bus\_route:

INSERT INTO bus\_route

VALUES

(189023, 13, '2021-11-12 10:00:00', '2021-18-12 10:00:00'),

(183623, 13, '2021-18-12 10:00:00', Null),

(178023, 7, '2021-21-12 15:15:00', Null),

(185023, 1, '2021-21-12 12:10:00', Null);

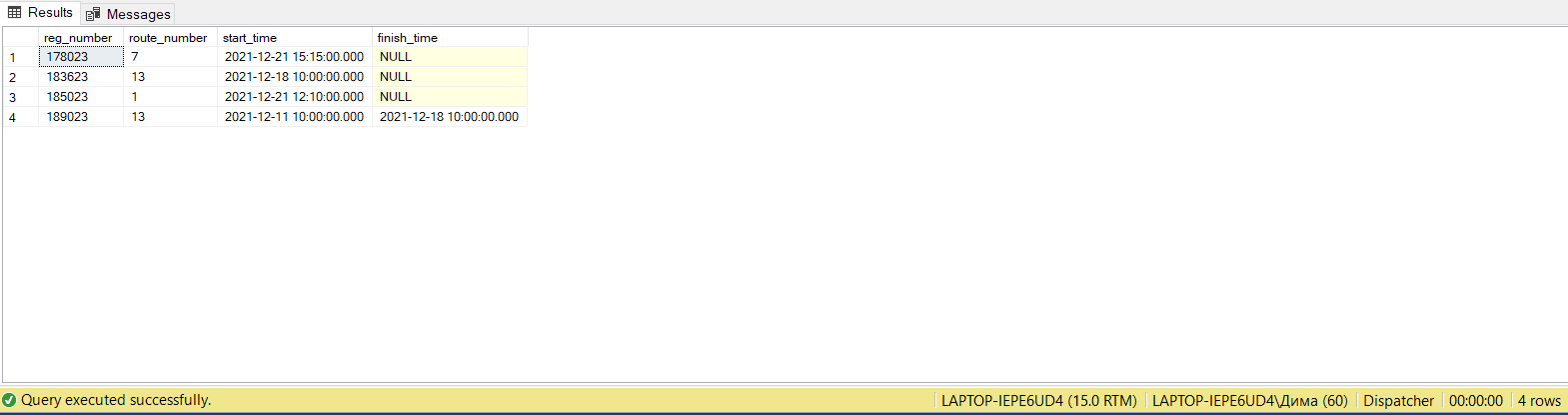


Рисунок 7 3– Результат заполнения таблицы bus\_route

Заполнение таблицы bus\_driver:

INSERT INTO bus\_driver

VALUES

(183623, 185, '2021-18-12 10:00:00', Null),

(178023, 2564, '2021-21-12 15:15:00', Null),

(185023, 5923, '2021-21-12 12:10:00', Null);

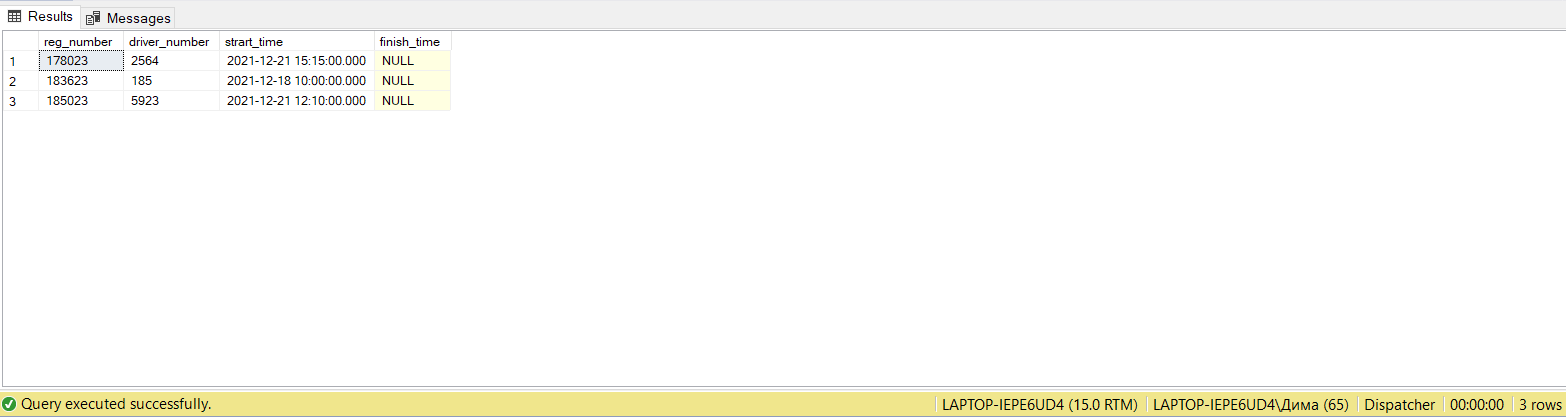


Рисунок 8 4– Результат заполнения таблицы bus\_driver

# Разработка объектов промежуточного слоя

## Представления

Cоздание представления ExperiencedDriversView:

CREATE VIEW ExperiencedDriversView AS

SELECT drivers.number AS Number,

drivers.surname AS Surname,

drivers.name AS Name,

drivers.patronymic AS Patronymic,

drivers.exp AS Experience,

drivers.note AS Note

FROM drivers

WHERE drivers.exp > 5

Данное представление выводит информацию о водителях, у которых стаж работы больше 5 лет:

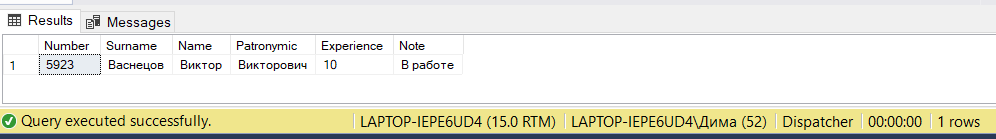


Рисунок 9 – Результат создания представления ExperiencedDriversView

Создание представления AppointmentsView:

CREATE VIEW RoutesView AS

SELECT

BR.route\_number AS RouteNumber,

BR.reg\_number AS BusNumber,

BD.driver\_number AS DriverNumber,

routes.start\_time AS StartTime,

routes.finish\_time AS FinishTime,

routes.interval AS Interval,

buses.state AS BusState,

drivers.note AS DriverNote

FROM

(SELECT route\_number, reg\_number FROM bus\_route

WHERE bus\_route.finish\_time IS NULL) AS BR

INNER JOIN routes ON BR.route\_number = routes.number

INNER JOIN buses ON BR.reg\_number = buses.reg\_number

INNER JOIN bus\_driver AS BD ON BR.reg\_number = BD.reg\_number

INNER JOIN drivers ON BD.driver\_number = drivers.number

Данное представление выводит полную информацию о текущих маршрутах (Номер маршрута; Номер автобуса; Табельный номер водителя; Время начала движения; Время окончания движения; Интервал движения; Состояние автобуса; Помечание водителя):

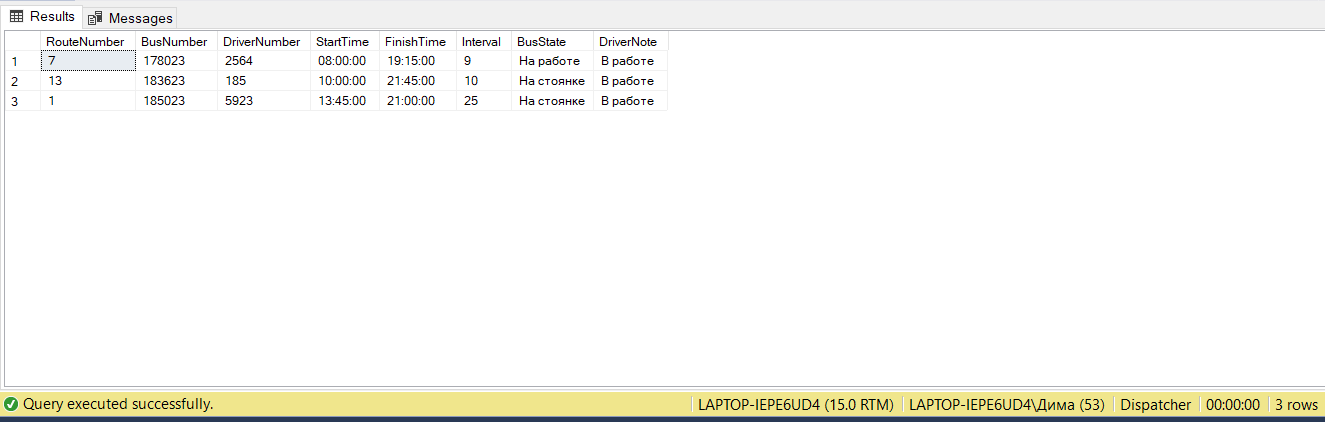


Рисунок 10 – Результат создания представления RoutesView

## Хранимые процедуры

Создание хранимой процедуры HireADriver. Эта процедура добавляет в таблицу bus\_driver новые данные о наёме водителя на выбранный автобус. При попытке поставить id, которое отсутствует в таблицах buses или drivers, произойдет исключение. Также, если автобус в ремонте или водитель на больничном, то также произойдет исключение:

GO

CREATE PROCEDURE HireADriver

@bus int,

@driver int

AS

BEGIN

IF (NOT EXISTS (SELECT reg\_number FROM buses Where reg\_number = @bus))

THROW 51000, 'Thes bus dont exist in the database', 1;

IF (NOT EXISTS (SELECT number FROM drivers Where number = @driver))

THROW 51000, 'Thes driver dont exist in the database', 1;

IF ((SELECT note FROM drivers Where number = @driver) = 'На больничном')

THROW 51000, 'Selected driver on sick leave', 1;

IF ((SELECT state FROM buses Where reg\_number = @bus) = 'В ремонте')

THROW 51000, 'The bus is under repair', 1;

INSERT INTO bus\_driver

VALUES (@bus, @driver, GETDATE(), Null)

END

Для проверки процедуры также добавим нового водителя и автобус:

INSERT INTO drivers

VALUES (1232, 'Петров', 'Валентин', 'Дядькович', 1, 'В работе');

INSERT INTO buses

VALUES (123230, 'Микроавтобус', 48, 'На стоянке', 'Исправен');

EXEC HireADriver 123230, 1232;

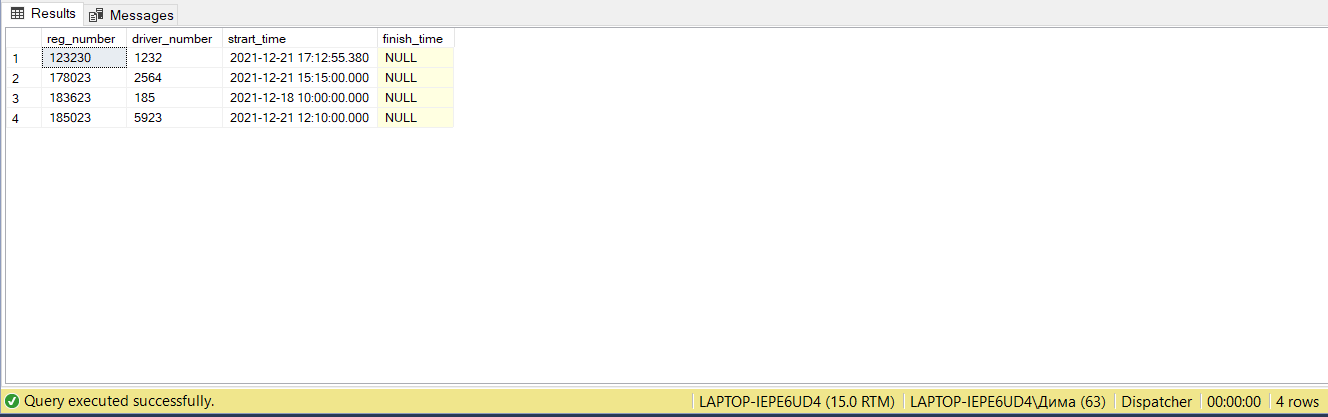


Рисунок 11 - Результат выполнения процедуры

Создание хранимой процедуры FireADriver. Эта процедура увольняет водителя с текущего автобуса (на котором он в данный момент работает), если текущего автобуса нет, то произойдет исключение:

GO

CREATE PROCEDURE FireADriver (@driver int)

AS

BEGIN

IF (NOT EXISTS (SELECT driver\_number FROM bus\_driver WHERE driver\_number = @driver AND finish\_time IS NULL))

THROW 51000, 'The driver does not work on any of the buses', 1;

UPDATE bus\_driver

SET finish\_time = GETDATE()

WHERE driver\_number = @driver AND finish\_time IS NULL

END

Вызов хранимой процедуры. Попробуем уволить водителя, нанятого в прошлой процедуре HireADriver:

EXEC FireADriver 1232;

SELECT \* FROM bus\_driver;

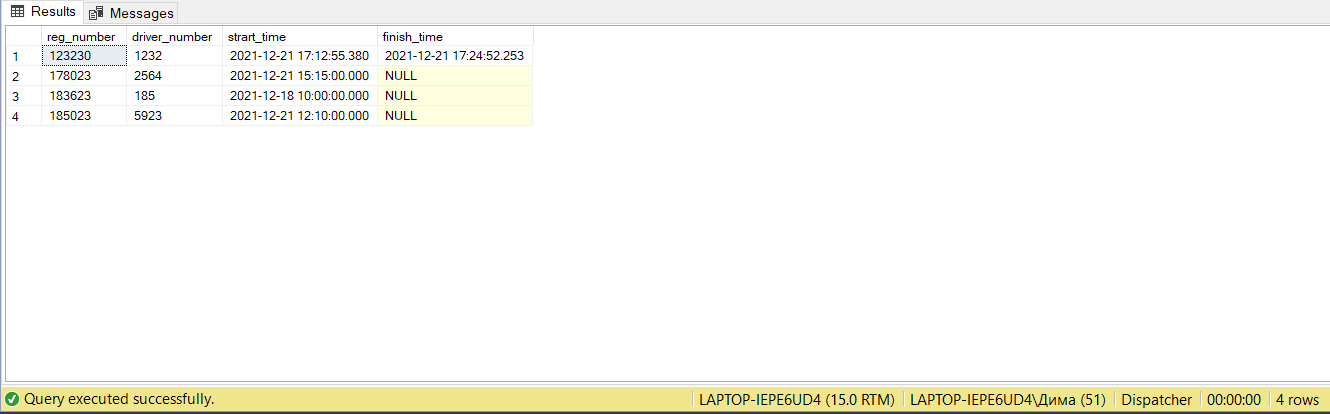


Рисунок 12 - Результат выполнения процедуры

## UDF

Создание определяемой пользователем функции, которая будет возвращать полную информацию о всех автобусах, работающих на выбранном маршруте:

CREATE FUNCTION BusesOnRoute (@route int)

RETURNS TABLE

AS

RETURN

SELECT

routes.number AS RouteNumber,

buses.reg\_number AS BusNumber,

buses.type AS BusType,

buses.state AS BusState,

buses.note AS BusNote

FROM (SELECT \* FROM bus\_route WHERE bus\_route.route\_number = @route AND bus\_route.finish\_time IS NULL) AS BR

INNER JOIN buses ON BR.reg\_number = buses.reg\_number

INNER JOIN routes ON BR.route\_number = routes.number

Вызов определяемой пользователем функции:

SELECT \* FROM BusesOnRoute (13)

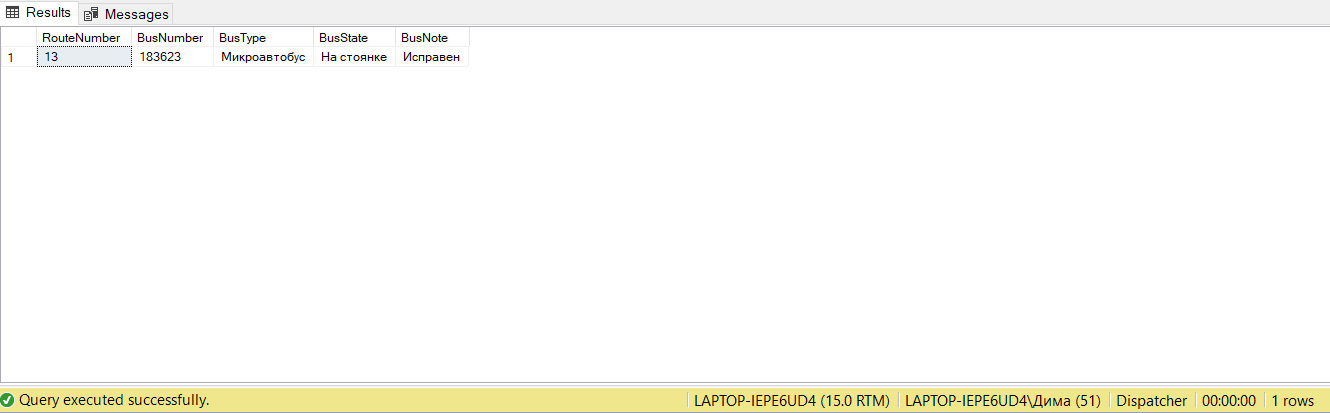


Рисунок 13 - Результат выполнения функции

Создание определяемой пользователем функции, которая будет возвращать информацию о списке всех водителей:

CREATE FUNCTION ListOfDrivers ()

RETURNS TABLE

AS

RETURN

SELECT \* FROM drivers

Вызов определяемой пользователем функции:

SELECT \* FROM ListOfDrivers();

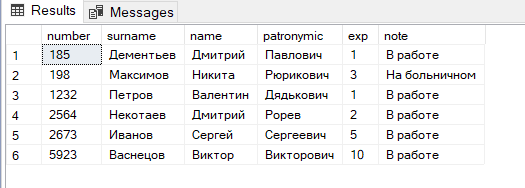


Рисунок 14 - Результат выполнения функции

Создание определяемой пользователем функции AvailableRoutes, которая будет возвращать список всех доступных маршрутов в настоящее время:

CREATE FUNCTION AvailableRoutes (@time time)

RETURNS TABLE

AS

RETURN

SELECT \* FROM routes WHERE @time BETWEEN routes.start\_time AND routes.finish\_time

Вызов хранимой функции:

SELECT \* FROM AvailableRoutes (GETDATE())

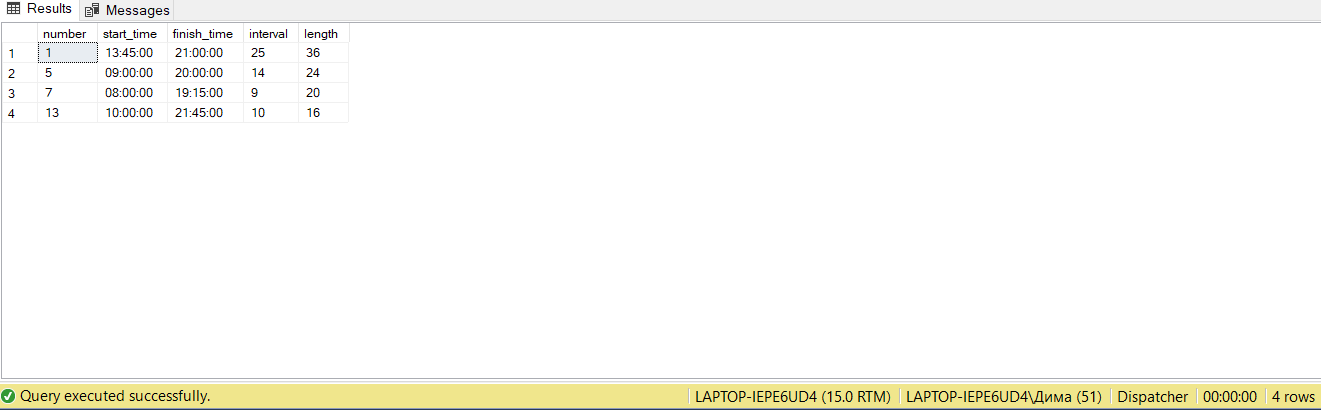


Рисунок 15 - Результат выполнения функции

# Резервное копирование

Использование стратегии резервного копирования базы данных и журнала транзакций.

Создание промежуточных резервных копий журнала транзакций, чтобы вести запись всех действий в базе данных, которые происходят между полными резервными копированиями базы данных. Таким образом, произойдет восстановление базы данных из самой последней полной резервной копии базы данных. Далее будут применены все резервные копии журнала транзакций, которые были созданы с момента последнего полного копирования.

Создание устройства резервного копирования DispatcherBackup и DispatcherBackupLog:

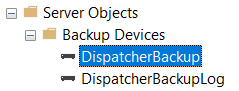


Рисунок 16 - Устройства резервного копирования

Теперь будет произведено полное резервное копирование базы данных Dispatcher в устройство DispatcherBackup и резервное копирование журнала в DispatcherBackupLog (рис. 16).

Создание резервного копирования:

BACKUP DATABASE Dispatcher

TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup\DispatcherBackup.bak'

WITH INIT

BACKUP LOG Dispatcher

TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL15.MSSQLSERVER\MSSQL\Backup\DispatcherBackupLog.bak'

WITH INIT

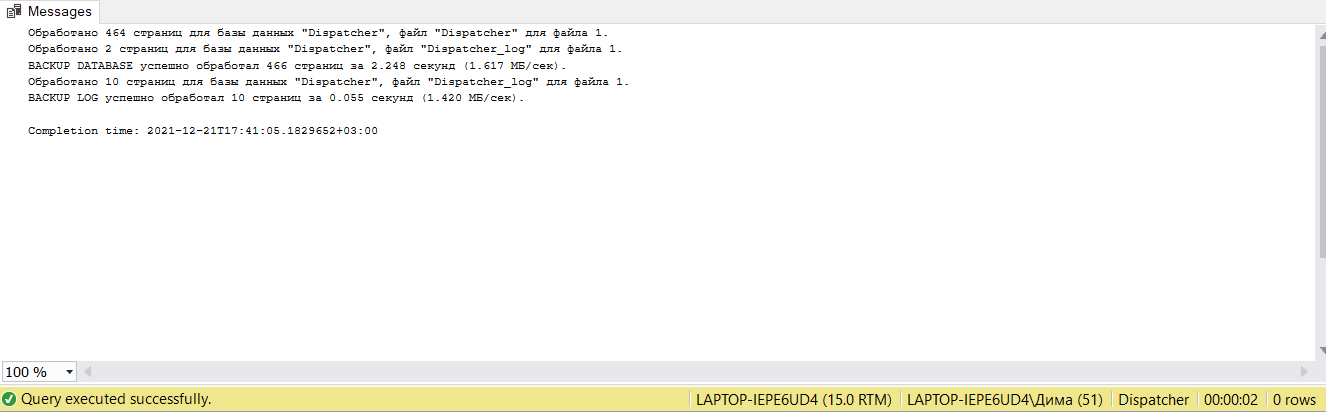


Рисунок 16 – Результат резервного копирования

# Диаграмма

На рис. 17 представлена диаграмма связей между таблицами получившейся базы данных.

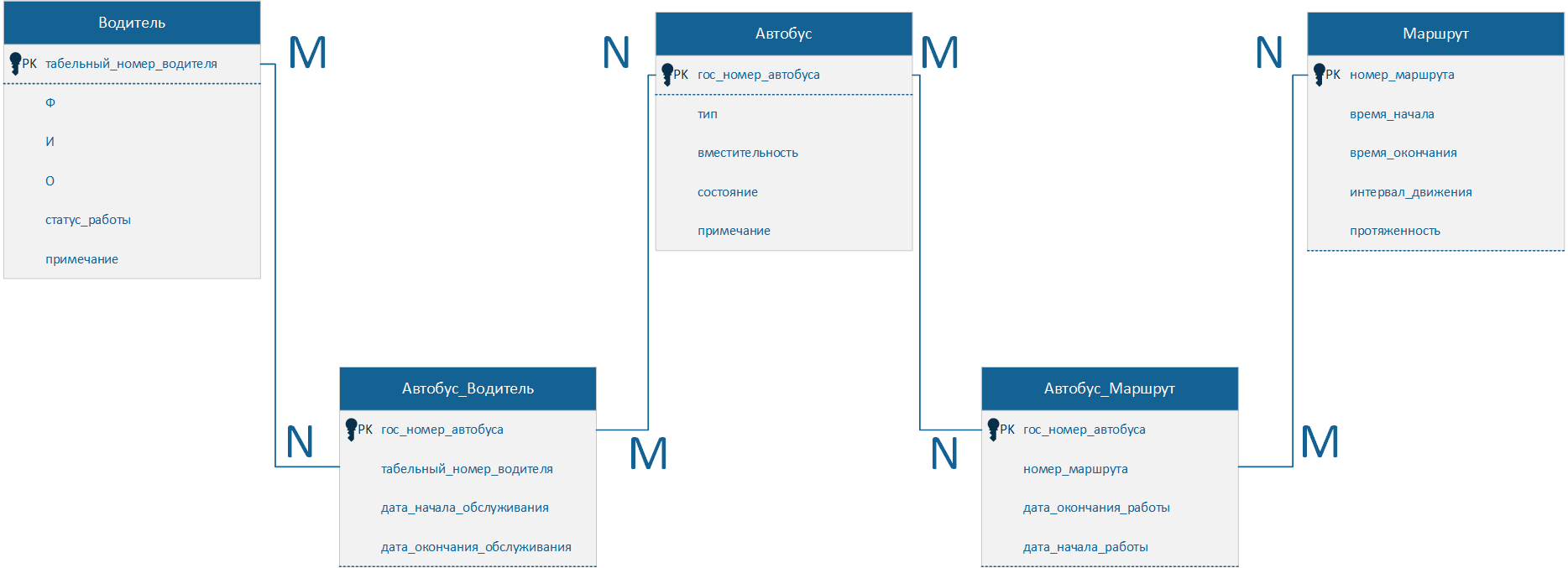


Рисунок 17 - Диаграмма связей между таблицами

# Выводы

В результате проделанного индивидуального домашнего задания были приобретены практические навыки проектирования базы данных. Были установлены связи между таблицами, созданы объекты промежуточного слоя, а также установлена стратегия резервного копирования.

**Список используемых источников**

1. Горячев А. В., Новакова Н. Е. Распределенные базы данных. Мет. указания к лаб. работам., СПб. Изд-во СПбГЭТУ «ЛЭТИ», 2008
2. Горячев А.В, Новакова Н.Е. Особенности разработки и администрирования приложений баз данных: учеб. пособие. СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2016. 68 с.
3. Дейт К. Введение в системы баз данных. : Пер. с англ. – 6-е изд. -К.:Диалектика, 1998.
4. Базы данных: модели, разработка, реализация: учеб./ Т.С.Карпова. - СПб. : Питер, 2001. - 303 с.
5. Базы данных: основы, проектирование, использование: учеб. пособие для вузов по направлению подгот. "Информатика и вычисл. техника" / М.П. Малыхина. - 2-е изд. - СПб. : БХВ-Петербург, 2007. - X, 517 с.
6. Базы данных. Теория и практика: учеб. для вузов по направлениям "Информатика и вычисл. техника" и "Информац. системы" / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М. : Высш. шк., 2005. - 463 с.