Содержание

[Введение 2](#_Toc478504154)

[Глава 1. Анализ предметной области 4](#_Toc478504155)

[1.1 Описание предметной области 4](#_Toc478504156)

[1.2 Описание программных продуктов, используемых при проектировании базы данных 4](#_Toc478504157)

[1.3 Создание структуры и нормализация базы данных 6](#_Toc478504158)

[Глава 2. Проектирование физической модели базы данных 10](#_Toc478504159)

[2.1 Проектирование БД 10](#_Toc478504160)

[2.2 Создание таблиц, индексов, триггеров, пользователей с различным уровнем доступа 11](#_Toc478504161)

[2.3 Заполнение таблиц данными 16](#_Toc478504162)

[2.4 Создание автоматических задач по резервному копированию всей базы и отдельных таблиц 17](#_Toc478504163)

[Глава 3. Создание запросов 21](#_Toc478504164)

[3.1 Описание языка T-SQL 21](#_Toc478504165)

[3.2 Создание запросов к базе данных 22](#_Toc478504166)

[3.3 Экспорт в Excel 23](#_Toc478504167)

[Заключение 29](#_Toc478504168)

[Список использованных источников 30](#_Toc478504169)

# Введение

Проектирование БД является очень важным этапом, от которого зависят последующие этапы разработки СУБД. Время, затраченное разработчиком на проектирование БД, обычно окупается высокой скоростью реализации проекта.

Перед созданием базы данных необходимо располагать описанием выбранной предметной области, которое должно охватывать реальные объекты и процессы, иметь всю необходимую информацию для удовлетворения предполагаемых запросов пользователя и определить потребности в обработке данных.

На основе такого описания на этапе проектирования базы данных осуществляется определение состава и структуры данных предметной области, которые должны находиться в базе данных и обеспечивать выполнение необходимых запросов и задач пользователя. Структура данных предметной области может отображаться информационно-логической моделью. На основе этой модели легко создается реляционная база данных.

Информационно-логическая модель отображает данные предметной области в виде совокупности информационных объектов и связей между ними. Эта модель представляет данные, подлежащие хранению в базе данных.

При разработке модели данных могут использоваться два подхода. В первом подходе сначала определяются основные задачи, для решения которых строится база, и выявляются потребности задач в данных. При втором подходе сразу устанавливаются типовые объекты предметной области. Наиболее рационально сочетание обоих подходов. Это связано с тем, что на начальном этапе, как правило, нет исчерпывающих сведений обо всех задачах.

Информационный объект — это информационное описание некоторой сущности — реального объекта, процесса, явления или события. Информационный объект образуется совокупностью логически взаимосвязанных реквизитов, представляющих качественные и количественные характеристики некоторой сущности предметной области.

Цель практических заданий – приобретение навыков анализа предметной области, проектирования базы данных, ее физической реализации в СУБД MS SQL Server.

Результат выполнения работы представляется в виде документа Word, который должен содержать:

* анализ предметной области;
* структуру спроектированных таблиц;
* схему данных со связями между таблицами.

# Глава 1. Анализ предметной области

## 1.1 Описание предметной области

Необходимо создать базу данных (БД) «Нагрузка преподавателей» с применением MS SQL Server 2008R2, которая должна содержать сведения о преподавателях, учебных дисциплинах, кафедрах вуза, преподавательском составе кафедр и распределении учебных дисциплин между преподавателями.

Под базой данных (БД) понимают совокупность хранящихся вместе данных при наличии такой минимальной избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений. Целью создания баз данных, как разновидности информационной технологии и формы хранения данных, является построение системы данных, не зависящих от принятых алгоритмов (программного обеспечения), применяемых технических средств и физического расположения данных в ЭВМ; обеспечивающих непротиворечивую и целостную информацию при не регламентируемых запросах. БД предполагает многоцелевое ее использование (несколько пользователей, множество форм документов и запросов одного пользователя).

## 1.2 Описание программных продуктов, используемых при проектировании базы данных

В ходе проектирования и разработки базы данных использовалась СУБД MS SQL Server. Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

Server 2008 Express — это бесплатная редакция системы SQL Server. Она идеально подходит для обучения работе с базами данных, для создания небольших серверных приложений и для распространения независимыми поставщиками ПО. В этот выпуск включен ряд мощных функциональных возможностей, в частности службы отчетов SQL Server 2008 — серверная платформа для создания и отправки классических и интерактивных отчетов — и графическая среда SQL Server 2008 Management Studio Express, упрощающая управление базами данных. (Права на свободное распространение для независимых разработчиков ПО)

SQL Server 2008 Workgroup представляет собой надежную платформу для управления данными и составления отчетов. Она позволяет безопасно выполнять удаленную синхронизацию и управлять приложениями в филиалах компании. В этот выпуск включены основные функции ядра базы данных SQL Server. Он легко обновляется до выпуска Standard или Enterprise. (Хранилище данных филиала, Создание отчетности в филиалах, Удаленная синхронизация).

SQL Server Standard — это законченная платформа для управления данными и выполнения операций бизнес-аналитики. Она демонстрирует лучшие в своем классе показатели простоты использования и степени управляемости приложений, на которых основана работа подразделений предприятия. (Поддержка приложений разной направленности, Поддержка систем принятия решений, требующих базовых возможностей по аналитике и созданию отчетов)

SQL Server 2008 Enterprise Edition — это комплексная платформа управления данными и бизнес-аналитики. Она обладает первоклассной масштабируемостью, возможностью создавать хранилища данных, продвинутыми средствами анализа и достаточной безопасностью, что позволяет использовать ее как основу для критически важных бизнес-приложений. Эта редакция позволяет консолидировать серверы и выполнять крупномасштабные OLTP-операции и создание отчетности. (OLTP (Online Transaction Processing) — обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы максимально быстрое время ответа.)

Выпуск SQL Server 2008 Developer позволяет разработчикам строить приложения любого вида на базе SQL Server. Этот выпуск включает все функциональные возможности выпуска SQL Server 2008 Enterprise, однако он лицензируется для разработки и тестирования системы, а не для применения в качестве рабочего сервера. Выпуск SQL Server 2008 Developer является идеальным выбором для тех, кто создает и тестирует приложения. Выпуск SQL Server 2008 Developer можно обновить для производственного использования.

SQL Server 2008 Web — это вариант с низкой общей стоимостью владения, предназначенный для размещения веб-узлов, который обеспечивает масштабируемость и функции управления для небольших и крупномасштабных веб-свойств.

SQL Server Compact — это встроенная база данных SQL Server, распространяемая бесплатно. Ее можно использовать для разработки автономных или мало связанных приложений для мобильных устройств, настольных компьютеров и веб-клиентов. SQL Server Compact работает под всеми версиями операционной системы Microsoft Windows, в том числе под Windows XP и Windows Vista, а также подходит для карманных ПК и смартфонов.

## 1.3 Создание структуры и нормализация базы данных

База данных должна содержать данные о преподавателях, занятиях и деканатов.

В таблице «Преподаватели» будет содержаться информация о ФИО, дата рождении, должности, стаже. Структуру таблицы «Преподаватели» можно посмотреть в таблице 1.1.

Таблица 1.1 – Структура таблицы «Преподаватели»

|  |  |
| --- | --- |
| Наименование | Тип данных |
| Код преподавателя | Числовой |
| ФИО | Строковый |
| Код учебной степени | Числовой |
| Дата рождения | Дата |
| Должность | Строковый |
| Стаж | Числовой |

В таблице «Тип занятий» будет содержаться информация о типе занятия и количестве часов. Структуру таблицы «Тип занятий» можно посмотреть в таблице 1.2.

Таблица 1.2 – Структура таблицы «Тип занятий»

|  |  |
| --- | --- |
| Наименование | Тип данных |
| Код занятий | Числовой |
| Тип занятий | Строковый |
| Количество часов | Числовой |

Таблица «Кафедра» будет содержать информацию о наименовании кафедр. Структуру таблицы «Кафедра» можно посмотреть в таблице 1.3.

Таблица 1.3 – Структура таблицы «Кафедра»

|  |  |
| --- | --- |
| Наименование | Тип данных |
| Код кафедры | Числовой |
| Наименование кафедры | Строковый |

Данные нужно привести к 3 НФ. Третья нормальная форма — одна из возможных нормальных форм отношения в реляционной базе данных.

Чтобы привести базу к третьей нормальной форме, надо:

1. Определить, в каких полях каких таблиц имеется взаимозависимость.

2. Создать соответствующие таблицы.

3. Создать или выделить первичные ключи. Каждая таблица должна иметь первичный ключ.

4. Создайте необходимые внешние ключи, которые образуют любое из отношений.

Для того, чтобы привести таблицы к 3 НФ. Таблица должна находится в 2 НФ и 1 НФ.

Переменная отношения находится в первой нормальной форме тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

В реляционной модели отношение всегда находится в первой нормальной форме по определению понятия отношение.

Так как таблицы уже находятся в первой нормальной форме, то приведём таблицы во 2 НФ. Переменная отношения находится во второй нормальной форме тогда и только тогда, когда она находится в первой нормальной форме и каждый не ключевой атрибут неприводимо зависит от её потенциального ключа.

Исходные таблицы находятся во 2 НФ. Для того, чтобы привести таблицу в 3 НФ необходимо создать несколько таблиц, для того, чтобы избавиться от избыточности данных и убрать повторения.

Необходимо добавить таблицу «Возраст преподавателей», где будет содержаться информация и преподавателях, их дате рождения и возрасте. Данные сведения редко используются и помещение их в отдельную таблицу избавит от избыточности и упростит работу оператора, не загружая его сведениями, которые не являются приоритетными. Структуру таблицы «Возраст преподавателей» можно посмотреть в таблице 1.4.

Таблица 1.4 – Структура таблицы «Возраст преподавателей»

|  |  |
| --- | --- |
| Наименование | Тип данных |
| Код возраста преподавателя | Числовой |
| Дата рождения | Дата |
| Возраст | Числовой |

Также можно добавить таблицу «Количество сотрудников по должностям», что увеличит удобство пользования базой данных, так как позволит вести более точный учет данных. Структуру таблицы «Количество сотрудников по должностям» можно посмотреть в таблице 1.5.

Таблица 1.5 – Структура таблицы «Количество сотрудников по должностям»

|  |  |
| --- | --- |
| Наименование | Тип данных |
| Код количества | Числовой |
| Должность | Строковый |
| Количество сотрудников | Числовой |

После того, как база данных приведена к 3 НФ можно приступить к проектированию базы данных.

# Глава 2. Проектирование физической модели базы данных

## 2.1 Проектирование БД

Физическая модель таблицы «Тип занятий» представлена в таблице 2.1.

Таблица 2.1 – Физическая модель таблицы «Тип занятий»

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| ZanId | Integer |
| TypeZan | String |
| CountHours | Integer |

Физическая модель таблицы «Преподаватели» представлена в таблице 2.2.

Таблица 2.2 – Физическая модель таблицы «Преподаватели»

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| PrepodId | Integer |
| FIO | String |
| UchebStepenId | Integer |
| DateBirth | Date |
| DolId | Integer |
| Stah | Integer |

Физическая модель таблицы «Возраст преподавателей» представлена в таблице 2.3.

Таблица 2.3 – Физическая модель таблицы «Возраст преподавателей»

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| PrepodId | Integer |
| DateBirth | Date |
| Age | Integer |

Физическая модель таблицы «Количество сотрудников» представлена в таблице 2.4.

Таблица 2.4 – Физическая модель таблицы «Количество сотрудников»

|  |  |
| --- | --- |
| Имя поля | Тип данных |
| CountSotrId | Integer |
| Dol | String |
| CountSotr | Integer |

## 2.2 Создание таблиц, индексов, триггеров, пользователей с различным уровнем доступа

Листинг 1 - Скрипт создания базы данных и таблиц

**USE** [master]

**GO**

*/\*\*\*\*\*\* Object: Database [dtl]*

**CREATE** **DATABASE** [dtl] **ON** **PRIMARY**

( NAME = N'dtl', FILENAME = N'c:**\P**rogram Files**\M**icrosoft SQL Server**\M**SSQL10\_50.MSSQLSERVER**\M**SSQL**\D**ATA**\d**tl.mdf' , **SIZE** = 3072KB , MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )

LOG **ON**

( NAME = N'dtl\_log', FILENAME = N'c:**\P**rogram Files**\M**icrosoft SQL Server**\M**SSQL10\_50.MSSQLSERVER**\M**SSQL**\D**ATA**\d**tl\_log.ldf' , **SIZE** =

1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)

**GO**

**ALTER** **DATABASE** [dtl] **SET** COMPATIBILITY\_LEVEL = 100

**GO**

**IF** (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))

**BEGIN**

**EXEC** [dtl].[dbo].[sp\_fulltext\_database] @action = 'enable'

**END**

**GO**

**ALTER** **DATABASE** [dtl] **SET** ANSI\_NULL\_DEFAULT OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** ANSI\_NULLS OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** ANSI\_PADDING OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** ANSI\_WARNINGS OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** ARITHABORT OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** AUTO\_CLOSE OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** AUTO\_CREATE\_STATISTICS **ON**

**GO**

**ALTER** **DATABASE** [dtl] **SET** AUTO\_SHRINK OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** AUTO\_UPDATE\_STATISTICS **ON**

**GO**

**ALTER** **DATABASE** [dtl] **SET** CURSOR\_CLOSE\_ON\_COMMIT OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** CURSOR\_DEFAULT GLOBAL

**GO**

**ALTER** **DATABASE** [dtl] **SET** CONCAT\_NULL\_YIELDS\_NULL OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** NUMERIC\_ROUNDABORT OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** QUOTED\_IDENTIFIER OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** RECURSIVE\_TRIGGERS OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** DISABLE\_BROKER

**GO**

**ALTER** **DATABASE** [dtl] **SET** AUTO\_UPDATE\_STATISTICS\_ASYNC OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** DATE\_CORRELATION\_OPTIMIZATION OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** TRUSTWORTHY OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** ALLOW\_SNAPSHOT\_ISOLATION OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** PARAMETERIZATION SIMPLE

**GO**

**ALTER** **DATABASE** [dtl] **SET** READ\_COMMITTED\_SNAPSHOT OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** HONOR\_BROKER\_PRIORITY OFF

**GO**

**ALTER** **DATABASE** [dtl] **SET** READ\_WRITE

**GO**

**ALTER** **DATABASE** [dtl] **SET** RECOVERY SIMPLE

**GO**

**ALTER** **DATABASE** [dtl] **SET** MULTI\_USER

**GO**

**ALTER** **DATABASE** [dtl] **SET** PAGE\_VERIFY CHECKSUM

**GO**

**ALTER** **DATABASE** [dtl] **SET** DB\_CHAINING OFF

**GO**

**USE** [dtl]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[Prepod] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[Prepod](

[PrepodId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[FIO] [nvarchar](50) **NULL**,

[UchebStepenId] [**INT**] **NOT** **NULL**,

[DateBirth] [**DATE**] **NULL**,

[DolId] [**INT**] **NOT** **NULL**,

[Stah] [**INT**] **NULL**,

**CONSTRAINT** [PK\_Prepod] **PRIMARY** **KEY** CLUSTERED

(

[PrepodId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[Group] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[**GROUP**](

[GroupId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[Name] [nvarchar](50) **NULL**,

**CONSTRAINT** [PK\_Group] **PRIMARY** **KEY** CLUSTERED

(

[GroupId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[ColSotr] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[ColSotr](

[CountSotrId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[Dol] [nvarchar](50) **NULL**,

[CountSotr] [**INT**] **NULL**,

**CONSTRAINT** [PK\_ColSotr] **PRIMARY** **KEY** CLUSTERED

(

[CountSotrId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[TypeZan] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[TypeZan](

[ZanId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[TypeZan] [nvarchar](50) **NULL**,

[CountHours] [**INT**] **NULL**,

**CONSTRAINT** [PK\_TypeZan] **PRIMARY** **KEY** CLUSTERED

(

[ZanId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[Zan] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[Zan](

[ZanId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[TypeZanId] [**INT**] **NULL**,

[Predmet] [nvarchar](50) **NULL**,

[GroupId] [**INT**] **NULL**,

[PrepodId] [**INT**] **NULL**,

[Semestr] [**INT**] **NULL**,

**CONSTRAINT** [PK\_Zan] **PRIMARY** **KEY** CLUSTERED

(

[ZanId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[VosrastPrepod] Script Date:*

*03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[VosrastPrepod](

[VozrastPrepodId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[PrepodId] [**INT**] **NOT** **NULL**,

[DateBirth] [**DATE**] **NULL**,

[Age] [**INT**] **NULL**,

**CONSTRAINT** [PK\_VosrastPrepod] **PRIMARY** **KEY** CLUSTERED

(

[VozrastPrepodId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: Table [dbo].[Student] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**SET** ANSI\_NULLS **ON**

**GO**

**SET** QUOTED\_IDENTIFIER **ON**

**GO**

**CREATE** **TABLE** [dbo].[Student](

[StudentId] [**INT**] **IDENTITY**(1,1) **NOT** **NULL**,

[GroupId] [**INT**] **NOT** **NULL**,

[FIO] [nvarchar](50) **NULL**,

[Address] [nvarchar](50) **NULL**,

[PhoneNumber] [nvarchar](50) **NULL**,

**CONSTRAINT** [PK\_Student] **PRIMARY** **KEY** CLUSTERED

(

[StudentId] **ASC**

)**WITH** (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = **ON**, ALLOW\_PAGE\_LOCKS = **ON**) **ON** [**PRIMARY**]

) **ON** [**PRIMARY**]

**GO**

*/\*\*\*\*\*\* Object: ForeignKey [FK\_Zan\_Group] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**ALTER** **TABLE** [dbo].[Zan] **WITH** **CHECK** **ADD** **CONSTRAINT** [FK\_Zan\_Group] **FOREIGN** **KEY**([GroupId])

**REFERENCES** [dbo].[**GROUP**] ([GroupId])

**ON** **UPDATE** CASCADE

**GO**

**ALTER** **TABLE** [dbo].[Zan] **CHECK** **CONSTRAINT** [FK\_Zan\_Group]

**GO**

*/\*\*\*\*\*\* Object: ForeignKey [FK\_Zan\_Prepod] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**ALTER** **TABLE** [dbo].[Zan] **WITH** **CHECK** **ADD** **CONSTRAINT** [FK\_Zan\_Prepod] **FOREIGN** **KEY**([PrepodId])

**REFERENCES** [dbo].[Prepod] ([PrepodId])

**ON** **UPDATE** CASCADE

**GO**

**ALTER** **TABLE** [dbo].[Zan] **CHECK** **CONSTRAINT** [FK\_Zan\_Prepod]

**GO**

*/\*\*\*\*\*\* Object: ForeignKey [FK\_Zan\_TypeZan] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**ALTER** **TABLE** [dbo].[Zan] **WITH** **CHECK** **ADD** **CONSTRAINT** [FK\_Zan\_TypeZan] **FOREIGN** **KEY**([TypeZanId])

**REFERENCES** [dbo].[TypeZan] ([ZanId])

**ON** **UPDATE** CASCADE

**GO**

**ALTER** **TABLE** [dbo].[Zan] **CHECK** **CONSTRAINT** [FK\_Zan\_TypeZan]

**GO**

*/\*\*\*\*\*\* Object: ForeignKey [FK\_VosrastPrepod\_Prepod] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**ALTER** **TABLE** [dbo].[VosrastPrepod] **WITH** **CHECK** **ADD** **CONSTRAINT** [FK\_VosrastPrepod\_Prepod] **FOREIGN** **KEY**([PrepodId])

**REFERENCES** [dbo].[Prepod] ([PrepodId])

**ON** **UPDATE** CASCADE

**ON** **DELETE** CASCADE

**GO**

**ALTER** **TABLE** [dbo].[VosrastPrepod] **CHECK** **CONSTRAINT** [FK\_VosrastPrepod\_Prepod]

**GO**

*/\*\*\*\*\*\* Object: ForeignKey [FK\_Student\_Group] Script Date: 03/28/2017 21:42:03 \*\*\*\*\*\*/*

**ALTER** **TABLE** [dbo].[Student] **WITH** **CHECK** **ADD** **CONSTRAINT** [FK\_Student\_Group] **FOREIGN** **KEY**([GroupId])

**REFERENCES** [dbo].[**GROUP**] ([GroupId])

**ON** **UPDATE** CASCADE

**GO**

**ALTER** **TABLE** [dbo].[Student] **CHECK** **CONSTRAINT** [FK\_Student\_Group]

**GO**

## 2.3 Заполнение таблиц данными

**INSERT** [dbo].[Prepod] ([PrepodId], [FIO], [UchebStepenId], [DateBirth], [DolId], [Stah]) **VALUES** (1, N'Иванов И.И.', 4, **CAST**(0x1C170B00 **AS** **DATE**), 2, 2)

**INSERT** [dbo].[**GROUP**] ([GroupId], [Name]) **VALUES** (1, N'11ПЗ-22')

**INSERT** [dbo].[**GROUP**] ([GroupId], [Name]) **VALUES** (2, N'12ПЗ-22')

**INSERT** [dbo].[ColSotr] ([CountSotrId], [Dol], [CountSotr]) **VALUES** (1, N'Преподаватели', 3)

**INSERT** [dbo].[ColSotr] ([CountSotrId], [Dol], [CountSotr]) **VALUES** (2, N'Лаборанты', 2)

**INSERT** [dbo].[TypeZan] ([ZanId], [TypeZan], [CountHours]) **VALUES** (1, N'лекция', 2)

**INSERT** [dbo].[TypeZan] ([ZanId], [TypeZan], [CountHours]) **VALUES** (2, N'практика', 2)

**INSERT** [dbo].[TypeZan] ([ZanId], [TypeZan], [CountHours]) **VALUES** (3, N'семинар', 4)

**INSERT** [dbo].[Zan] ([ZanId], [TypeZanId], [Predmet], [GroupId], [PrepodId], [Semestr]) **VALUES** (1, 1, N'физика', 1, 1, 1)

**INSERT** [dbo].[Zan] ([ZanId], [TypeZanId], [Predmet], [GroupId], [PrepodId], [Semestr]) **VALUES** (2, 1, N'физика', 1, 1, 2)

**INSERT** [dbo].[Student] ([StudentId], [GroupId], [FIO], [Address], [PhoneNumber]) **VALUES** (1, 1, N'Иванов И.И.', N'Чкалова, 22', N'12-12-12')

**INSERT** [dbo].[Student] ([StudentId], [GroupId], [FIO], [Address], [PhoneNumber]) **VALUES** (2, 2, N'Петров И.И.', N'Строителей, 3', N'21-32-12')

**INSERT** [dbo].[Student] ([StudentId], [GroupId], [FIO], [Address], [PhoneNumber]) **VALUES** (3, 1, N'Сидоров И.С.', N'Московский, 3', N'23-21-34')

**INSERT** [dbo].[Student] ([StudentId], [GroupId], [FIO], [Address], [PhoneNumber]) **VALUES** (4, 1, N'Богданович Ю.Г.', N'Чкалова, 3', N'234-23-23')

## 2.4 Создание автоматических задач по резервному копированию всей базы и отдельных таблиц

Для того, чтобы SQL Server автоматически создавал резервные копии базы данных необходимо нажать ПКМ по названию базы данных в обозревателе объектом, затем выбрать пункт «Задачи», а затем «Создать резервную копию». Откроется окно, представленное на рисунке 2.1.

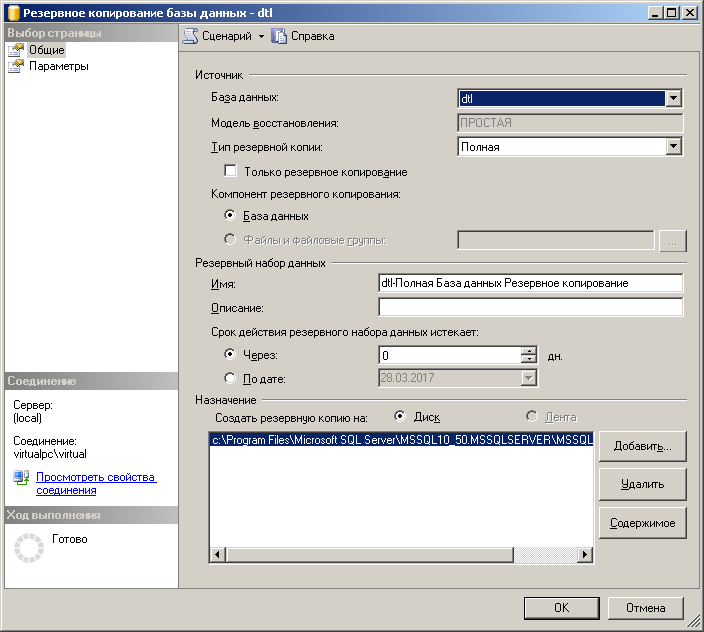


Рисунок 2.1 – Создание резервной копии базы данных

Теперь необходимо произвести настройки базы данных, установить интервал времени, через которое будет создаваться копия, тип резервной копии, куда будет сохраняться.

Выбранные настройки представлены на рисунке 2.2.

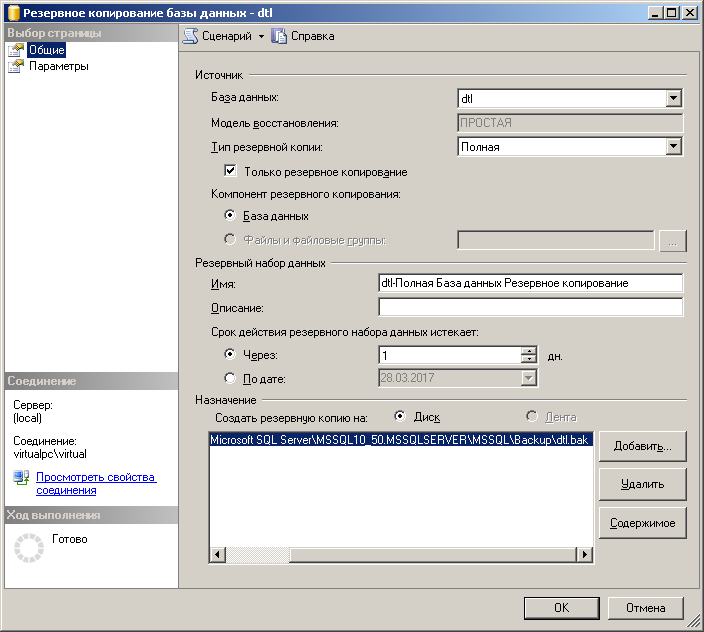


Рисунок 2.2 – Выбранные настройки

Параметры создания резервной копии представлены на рисунке 2.3.

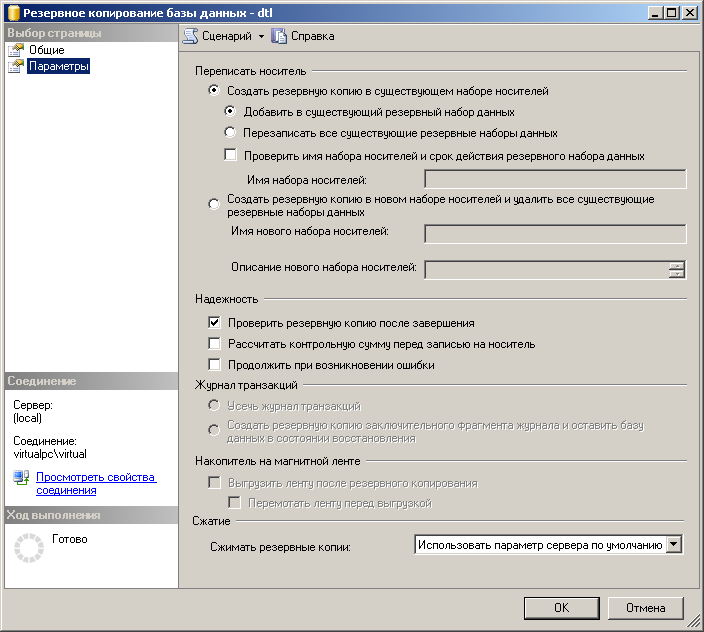


Рисунок 2.3 – Параметры настройки создания резервной копии базы данных

# Глава 3. Создание запросов

## 3.1 Описание языка T-SQL

Transact-SQL (T-SQL) — процедурное расширение языка SQL, созданное компанией Microsoft (для Microsoft SQL Server) и Sybase (для Sybase ASE).

SQL был расширен такими дополнительными возможностями как:

* управляющие операторы,
* локальные и глобальные переменные,
* различные дополнительные функции для обработки строк, дат, математики и т. п.,
* поддержка аутентификации Microsoft Windows.

Язык Transact-SQL является ключом к использованию MS SQL Server. Все приложения, взаимодействующие с экземпляром MS SQL Server, независимо от их реализации и пользовательского интерфейса, отправляют серверу инструкции Transact-SQL.

Директивы сценария — это специфические команды, которые используются только в MS SQL. Эти команды помогают серверу определять правила работы со скриптом и транзакциями. Типичные представители: GO — информирует программы SQL Server об окончании пакета инструкций Transact-SQL, EXEC (или EXECUTE) — выполняет процедуру или скалярную функцию.

В Transact-SQL существуют специальные команды, которые позволяют управлять потоком выполнения сценария, прерывая его или направляя в нужную ветвь.

Блок группировки — структура, объединяющая список выражений в один логический блок (BEGIN … END).

Блок условия — структура, проверяющая выполнения определённого условия (IF … ELSE).

Блок цикла — структура, организующая повторение выполнения логического блока (WHILE … BREAK … CONTINUE).

Переход — команда, выполняющая переход потока выполнения сценария на указанную метку (GOTO).

Задержка — команда, задерживающая выполнение сценария (WAITFOR).

Вызов ошибки — команда, генерирующая ошибку выполнения сценария (RAISERROR).

## 3.2 Создание запросов к базе данных

На рисунке 3.1 представлен запрос поиска данных

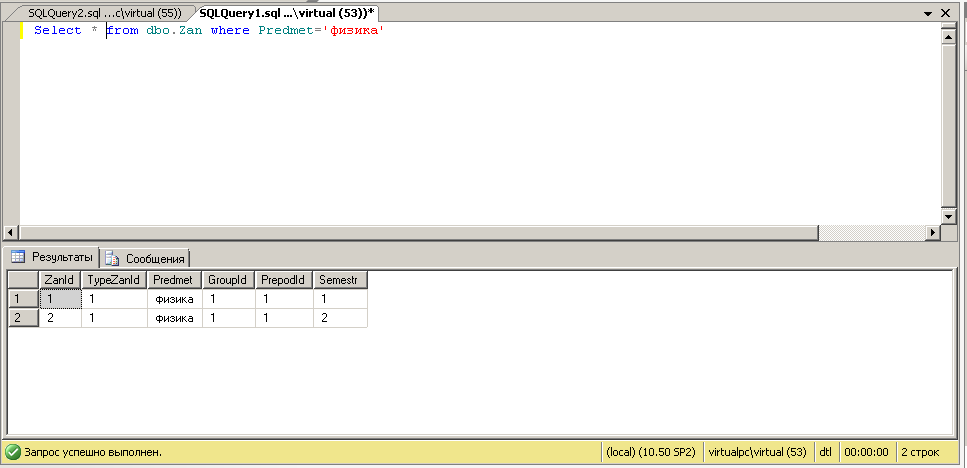


Рисунок 3.1 – Запрос к базе данных

На рисунке 3.2 представлен запрос поиска предмета и преподавателя.

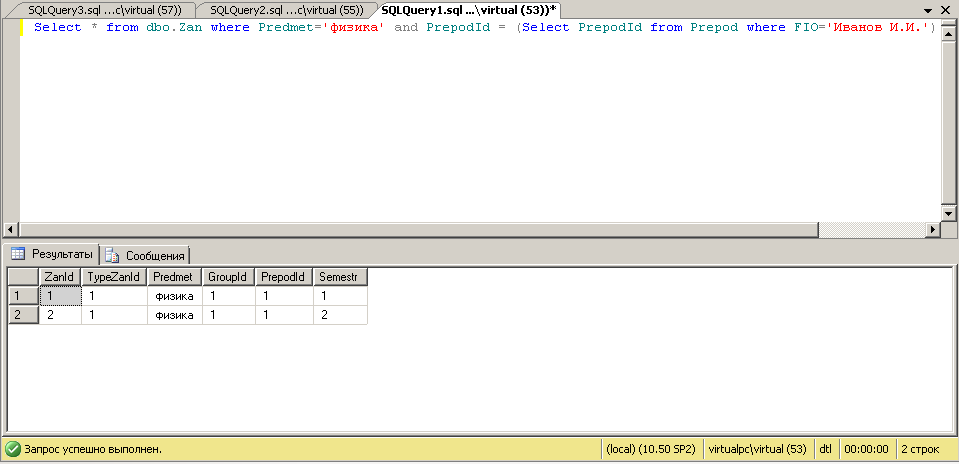


Рисунок 3.2 – Запрос к базе данных

## 3.3 Экспорт в Excel

Для экспорта данных в Excel необходимо выбрать настройки проекта (рис. 3.3).

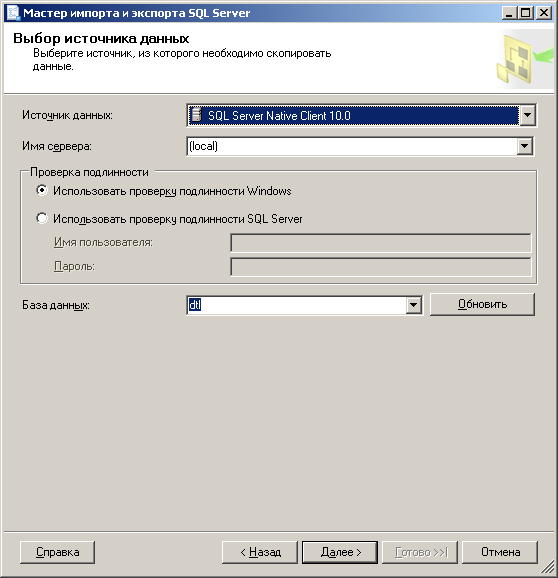


Рисунок 3.3 – Параметры проекта

На рисунке 3.4 представлена настройка Excel-документа для экспорта.

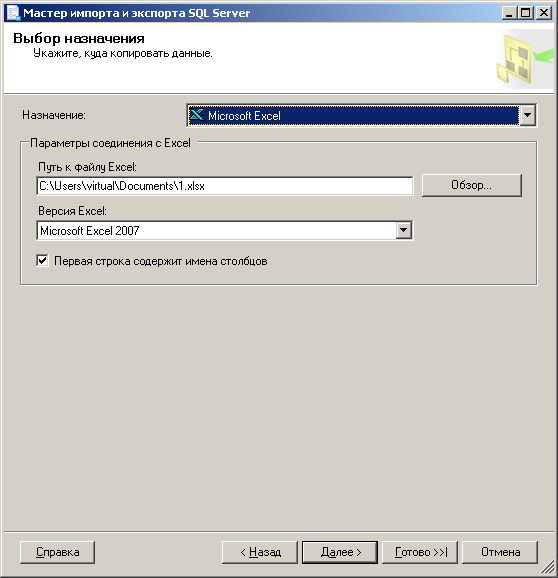


Рисунок 3.4 – Настройка файла экспорта

На рисунке 3.5 представлены параметры экспорта.

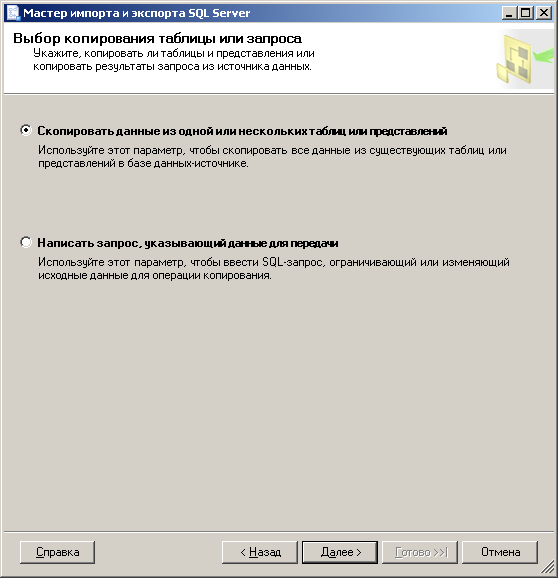


Рисунок 3.5 – Параметры экспорта

На рисунке 3.6 представлен список таблиц для экспорта.

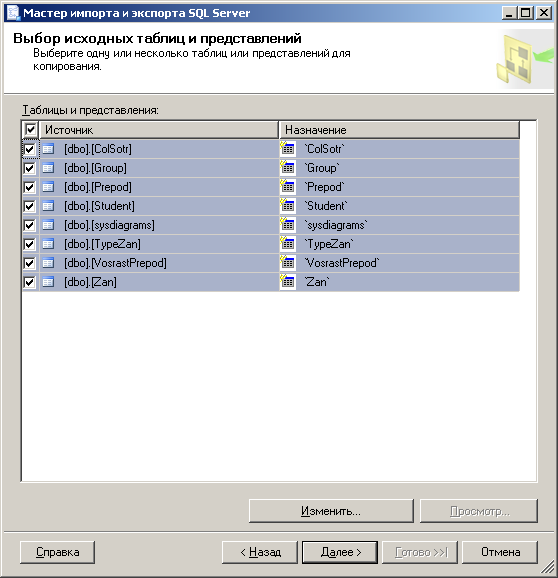


Рисунок 3.6 – Список таблиц для экспорта

На рисунке 3.7 представлены основные ошибки. Которые могут возникнуть в результате экспорта данных.

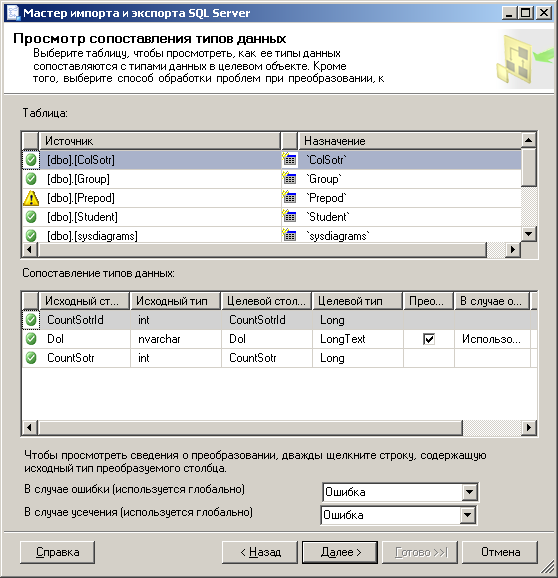


Рисунок 3.7 – Возможные ошибки экспорта данных

# Заключение

В ходе выполнения работы были исследованы существующие подходы и подходы распределения учебной нагрузки преподавателей кафедры университета.

В подходе предоставляется возможность ручного распределения учебной нагрузки. Ручной вариант заключается в следующем: система, каждому преподавателю присваивает определенное количество дисциплин по всем дисциплинам данной кафедры. При этом система учитывает должность и количество часов, отведенных для текущего преподавателя, и выбирает именно те дисциплины, в которых суммарное количество часов, которые она занимает, совпадает с нормой часов для текущего преподавателя. В случае неудачного распределения учебной нагрузки преподавателей происходит процесс перераспределения с учетом прошлых неудачных результатов.

Такой подход позволяет уменьшить затраты времени на распределение учебной нагрузки и уменьшить количество ошибок, возникающих при распределении учебной нагрузки вручную без помощи системы.

После анализа этого метода были выявлены следующие преимущества:

* метод стимулирует преподавания дисциплин в одном семестре, поскольку это, не влияя на величину штата ПВС, приводит к уменьшению учебной нагрузки;
* алгоритмы распределения, на которых базируется расчет, мало зависят от соотношения часов аудиторной и самостоятельной работы студента, следовательно, при увеличении СРС штат кафедр практически не изменится;
* становится невыгодным искусственное формирование мелких потоков, а также формирование неполных академических групп.

В результате анализа можно сделать вывод, что внедрение методов из расчета штатного расписания, учебная часть будет больше эффективность работы путем уменьшения затрат времени на выполнение рутинных операций расчета штатного расписания.

# Список использованных источников

1. Аткинсон, Леон MySQL. Библиотека профессионала; М.: Вильямс, 2010. - 624 c.
2. Бек, Кент Шаблоны реализации корпоративных приложений; М.: Вильямс, 2008. - 369 c.
3. Веймаер, Р.; Сотел, Р. Освой самостоятельно Microsoft SQL Server 2000 за 21 день (+ CD-ROM); М.: Вильямс, 2013. - 549 c.
4. Гандерлой, Майк; Харкинз, Сьюзан Сейлз Автоматизация Microsoft Access с помощью VBA; М.: Вильямс, 2013. - 416 c.
5. Гетц, Кен; Джинберт, Майкл; Литвин, Пол Access 2000. Руководство разработчика. Том 1. Настольные приложения. том 1; Киев: BHV, 2008. - 576 c.
6. Голицына, О.Л. и др. Базы данных; Форум; Инфра-М, 2013. - 399 c.
7. Гринченко, Н.Н. и др. Проектирование баз данных. СУБД Microsoft Access; Горячая Линия Телеком, 2012. - 613 c.
8. Дейт, К.Дж. Введение в системы баз данных; К.: Диалектика; Издание 6-е, 2012. - 360 c.
9. Дэвидсон, Луис проектирование баз данных на SQL Server 2000; Бином, 2009. - 631 c.
10. Дюваль, Поль М. Непрерывная интеграция. Улучшение качества программного обеспечения и снижение риска; М.: Вильямс, 2008. - 497 c.
11. Каратыгин, С.; Тихонов, А. Работа в Paradox для Windows 5.0 на примерах; М.: Бином, 2011. - 512 c.
12. Каратыгин, Сергей Access 2000 на примерах. Руководство пользователя с примерами; М.: Лаборатория Базовых Знаний, 2012. - 376 c.
13. Кауфельд, Джон Microsoft Office Access 2003 для "чайников"; М.: Диалектика, 2013. - 439 c.
14. Каучмэн, Джейсон; Швинн, Ульрике Oracle 8i CertifiedProfessionaql DBA Подготовка администраторов баз данных; ЛОРИ, 2009. - 510 c.
15. Луни, Кевин; Брила, Боб Oracle 10g. Настольная книга администратора баз данных; М.: Лори, 2008. - 365 c.
16. Мак-Федрис, Пол Формы, отчеты и запросы в Microsoft Access 2003; М.: Вильямс, 2010. - 416 c.
17. Наумов, А.Н.; Вендров, А.М.; Иванов, В.К. и др. Системы управления базами данных и знаний; М.: Финансы и статистика, 2010. - 352 c.
18. Нимик, Ричард Дж Oracle9i. Оптимизация производительности. Советы и методы; М.: Лори, 2012. - 648 c.
19. Озкарахан, Э. Машины баз данных и управление базами данных; М.: Мир, 2009. - 551 c.
20. Постолит, Анатолий Visual Studio .NET: разработка приложений баз данных; СПб: БХВ, 2009. - 544 c.
21. Редько, В.Н.; Бассараб, И.А. Базы данных и информационные системы; Знание, 2011. - 602 c.
22. Тимошок, Т.В. Microsoft Office Access 2007: самоучитель; Вильямс, 2008. - 464 c.
23. Тоу, Дэн Настройка SQL; СПб: Питер, 2009. - 539 c.