

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Шебзухова Татьяна Александровна

Должность: Директор Пятигорского института (филиал) Северо-Кавказского
федерального университета

Дата подписания: 27.05.2025 16:27:15

Уникальный программный ключ:

d74ce93cd40e39275c3ba2f58486412a1c8ef96f

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное образовательное учреждение
высшего образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Пятигорский институт (филиал) СКФУ

Колледж Пятигорского института (филиал) СКФУ

УТВЕРЖДАЮ

**Директор Пятигорского института
(филиал) СКФУ**

Т.А.Шебзухова

**ПМ.07 СОАДМИНИСТРИРОВАНИЕ БАЗ ДАННЫХ И СЕРВЕРОВ
ПМ.07.01 УПРАВЛЕНИЕ И АВТОМАТИЗАЦИЯ БАЗ ДАННЫХ
МЕТОДИЧЕСКИЕ УКАЗАНИЯ ДЛЯ ВЫПОЛНЕНИЯ ПРАКТИЧЕСКИХ
ЗАНЯТИЙ**

Специальности СПО

09.02.07 Информационные системы и программирование

Пятигорск 2025

Методические указания для выполнения практических занятия по дисциплине ПМ.07.01 Управление и автоматизация баз данных составлены в соответствии с требованиями ФГОС СПО. Предназначены для студентов, обучающихся по специальности 09.02.07 Информационные системы и программирование.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

При изучении предмета следует соблюдать единство терминологии и обозначения в соответствии с действующими стандартами, Международной системой единицы (СИ).

В результате изучения Управления и автоматизация баз данных студенты *должны*:

знать:

модели данных, основные операции и ограничения;
технологии установки и настройки сервера баз данных;
требования к безопасности сервера базы данных;
государственные стандарты и требования к обслуживанию баз данных.

уметь:

проектировать и создавать базы данных;
выполнять запросы по обработке данных на языке SQL;
осуществлять основные функции по администрированию баз данных;
разрабатывать политику безопасности SQL сервера, базы данных и отдельных объектов базы данных;
владеть технологиями проведения сертификации программного средства.

иметь практический опыт в:

участии в соадминистрировании серверов;
разработке политики безопасности SQL сервера, базы данных и отдельных объектов базы данных;
применении законодательства Российской Федерации в области сертификации программных средств информационных технологий.

В результате освоения учебной дисциплины студент должен овладевать:

Общими компетенциями:

ОК 01. Выбирать способы решения задач профессиональной деятельности, применительно к различным контекстам.

ОК 02. Использовать современные средства поиска, анализа и интерпретации информации и информационные технологии для выполнения задач профессиональной деятельности.

ОК 03. Планировать и реализовывать собственное профессиональное и личностное развитие.

ОК 04. Эффективно взаимодействовать и работать в коллективе и команде.

ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке с учетом особенностей социального и культурного контекста.

ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях.

ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

Профессиональными компетенциями:

ПК 7.1. Выявлять технические проблемы, возникающие в процессе эксплуатации баз данных и серверов.

ПК 7.2. Осуществлять администрирование отдельных компонент серверов.

ПК 7.3. Формировать требования к конфигурации локальных компьютерных сетей и серверного оборудования, необходимые для работы баз данных и серверов.

ОБЩИЕ МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ ПРАКТИЧЕСКИХ РАБОТ

По дисциплине Управление и автоматизация баз данных практические работы содержат задачи и теоретические вопросы. Варианты для каждого обучающегося - индивидуальные. Задачи и ответы на вопросы, выполненные не по своему варианту, не засчитываются.

Практическая работа выполняется в отдельной тетради. Условия задачи и формулировки вопросов переписываются полностью. Формулы, расчеты, ответы на вопросы пишутся ручкой, а чертежи, схемы и рисунки выполняются карандашом, на графиках и диаграммах указывается масштаб. Вначале задача решается в общем виде, затем делаются расчёты по условию задания. Решение задач обязательно ведется в Международной системе единиц (СИ).

При выполнении практической работы необходимо следовать методическим указаниям: повторить краткое содержание теории, запомнить основные формулы и законы, проанализировать пример выполнения аналогичного задания, затем преступить непосредственно к решению задачи. К зачету допускаются студенты, получившие положительные оценки по всем практическим работам.

Правила выполнения практических работ.

1. Студент должен прийти на практическое занятие подготовленным к выполнению практической работы.
2. Каждый студент после проведения работы должен представить отчет о проделанной работе с анализом полученных результатов и выводом по работе.
3. Таблицы и рисунки следует выполнять с помощью чертежных инструментов (линейки, циркуля, и т.д.) карандашом с соблюдением ЕСКД.
4. Расчет следует проводить с точностью до двух значащих цифр.
5. Исправления проводить на обратной стороне листа. При мелких исправлениях неправильное слово (буква, число и т.п.) аккуратно зачеркивается и над ним пишут правильное пропущенное слово (букву, число и т.п.).
6. Вспомогательные расчеты можно выполнять на отдельных листах, а при необходимости на листах отчета.
7. Если студент не выполнит практическую работу или часть работы, то он выполнит ее во внеурочное время, согласованное с преподавателем.
8. Оценку по практической работе студент получает с учетом срока выполнения работы, если;
 - расчеты выполнены правильно и в полном объеме;
 - сделан анализ проделанной работы и вывод по результатам работы;
 - студент может пояснить выполнение любого этапа работы;
 - отчет выполнен в соответствии с требованиями к выполнению работы.

Практическая работа №1.

Простые отчеты.

Изучение назначения отчёта и его структуры (разделов, источника данных, элементов управления), добавления в отчет сортировки, группировки данных вычисляемых полей, а также практика формирования отчётов различной сложности с помощью режимов конструктора, мастеров отчетов и автоотчетов — цель данного задания. Работа рассчитана на 6 часов аудиторных занятий и 4 часа самостоятельной работы.

Порядок выполнения практической работы

1. Построить простейший отчет на основе данных одной таблицы.
2. Создать отчет по одной таблице с группировкой по значению поля.

Практическая работа №2.

Сложные отчеты

Изучение назначения отчёта и его структуры (разделов, источника данных, элементов управления), добавления в отчет сортировки, группировки данных вычисляемых полей, а также практика формирования отчётов различной сложности с помощью режимов конструктора, мастеров отчетов и автоотчетов — цель данного задания. Работа рассчитана на 6 часов аудиторных занятий и 4 часа самостоятельной работы.

Порядок выполнения практической работы

1. Организовать отчета по нескольким таблицам одной базы данных на основе SQL-запроса.
2. Сформировать отчет по нескольким таблицам на основе SQL-запроса группировкой по значению поля.

Практическая работа №3

Отчеты с вычисляемыми полями.

Изучение назначения отчёта и его структуры (разделов, источника данных, элементов управления), добавления в отчет сортировки, группировки данных вычисляемых полей, а также практика формирования отчётов различной сложности с помощью режимов конструктора, мастеров отчетов и автоотчетов — цель данного задания. Задание рассчитано на 6 часов аудиторных занятий и 4 часа самостоятельной работы.

Порядок выполнения практической работы

1. Добавить в отчет вычисляемые поля.
 2. Ответить на один из контрольных вопросов к заданию.
- Результатом выполнения заданий являются пять созданных отчетов.

Вариант 1. Отчеты к базе данных «Студенческая научная работа»

1. Для создания отчета использовать таблицу **Студенты**. В шапку отчета внести текст: «Список студентов». В отчет вывести поля: *Номер зачетной книжки, Фамилия, Имя, Отчество, Группа, Дата рождения*. Строки отчета упорядочить по полю *Номер зачетной книжки*. Отчет сохранить под именем **Список студентов**.

2. В отчете **Список студентов** сделать группировку по студенческой группе. Для каждой образованной группы строк выдать новый заголовок «Студенческая группа Группа». Внутри группы строки упорядочить по полю *Номер зачетной книжки*. Созданный отчет сохранить под именем **Список студентов по группам**.

3. Для создания отчета использовать таблицы **Студенты, Научные руководители, Кафедры**. В шапку отчета внести текст: «Специализация студентов». В отчет вывести поля: *Номер зачетной книжки, Фамилия, Имя, Отчество, Группа студента, Фамилия, Имя, Отчество* научного руководителя, *Код кафедры, Название кафедры*. Строки отчета упорядочить по полю *Номер зачетной книжки*. Отчет сохранить под именем **Специализация**.

4. В отчете **Специализация** сделать группировку по полю *Код кафедры*. Для каждой образованной группы строк выдать новый заголовок «Кафедра Название кафедры». Внутри кафедры строки упорядочить по полям *Фамилия, Имя, Отчество* научного руководителя. Сформированный отчет сохранить под именем **Специализация по кафедрам**.

5. В отчете **Специализация по кафедрам** предусмотреть для каждой кафедры вычисление общего числа студентов, специализирующихся на ней, общего числа преподавателей, являющихся научными руководителями. Новому отчету дать имя **Специализация по кафедрам с итогами**.

Вариант 2. Отчеты к базе данных «Абоненты телефонной компании»

1. Для создания отчета использовать таблицу **Абоненты**. В шапку отчета внести текст: «Список абонентов». В отчет вывести поля: *Регистрационный номер абонента, Фамилия, Имя, Отчество, Улица, Дом, Квартира*. Строки отчета упорядочить по полю *Регистрационный номер абонента*. Отчет сохранить под именем **Список абонентов**.

2. В отчете **Список абонентов** сделать группировку по полю *Улица*. Для каждой образованной группы строк выдать новый заголовок «Улица Улица». Внутри группы строки упорядочить по полю *Регистрационный номер абонента*. Созданный отчет сохранить под именем **Список абонентов по улицам**.

3. Для создания отчета использовать таблицы **Абоненты, Телефоны, Оплата**. В шапку отчета внести текст: «Текущая оплата за телефон». В отчет вывести поля: *Регистрационный номер абонента, Фамилия, Имя, Отчество, Улица, Дом, Квартира, Номер телефона, Тариф оплаты за телефон, Сумма оплаты за текущий месяц*. Строки отчета упорядочить по полю *Регистрационный номер абонента*. Отчет сохранить под именем **Текущая оплата**.

4. В таблицу **Телефоны** добавить новое поле *Категория* (тип текстовый, длина 15). Заполнить это поле, используя такие виды категорий абонента: без льгот, пенсионер, ветеран, инвалид, многодетный и т.д. В отчет **Текущая оплата** внести поле *Категория* и сделать по нему группировку. Для каждой образованной группы строк выдать новый заголовок «Категория Категория». Внутри группы строки упорядочить по полю *Регистрационный номер абонента*. Сформированный отчет сохранить под именем **Текущая оплата по категориям**.

5. В отчете **Текущая оплата по категориям** предусмотреть для всякой категории вычисление общего числа абонентов, входящих в нее, и итоговой суммы недоплаты в текущем месяце. Новому отчету дать имя **Текущая оплата по категориям с итогами**.

Вариант 3. Отчеты к базе данных «Статистика медицинского учреждения»

1. Для создания отчета использовать таблицу **Пациенты**. В шапку отчета внести текст: «Список пациентов». В отчет вывести поля: *Номер карты, Фамилия, Имя, Отчество, Улица, Дом, Квартира, Дата поступления*. Строки отчета упорядочить по полю *Номер карты*. Отчет сохранить под именем **Список пациентов**.

2. В отчете **Список пациентов** сделать группировку по полю *Дата поступления*. Для каждой образованной группы строк выдать новый заголовок «Дата Дата поступления».

Все строки с одной и той же датой упорядочить по полю *Номер карты*. Созданный отчет сохранить под именем **Список пациентов по дате поступления**.

3. Для создания отчета использовать таблицы **Пациенты, Диагнозы, Отделения**. Предварительно в таблицу **Пациенты** добавить и заполнить новое поле *Дата фактической выписки* (тип Дата / Время, короткий формат). В шапку отчета внести текст: «Карты пациентов». В отчет вывести поля: *Номер карты, Фамилия, Имя, Отчество, Улица, Дом, Квартира, Ребенок / Взрослый, Диагноз, Код отделения, Название отделения, Дата поступления, Нормативное количество дней для стационарного лечения, Дата фактической выписки*. Строки отчета упорядочить по полю *Номер карты*. Отчет сохранить под именем **Карты пациентов**.

4. В отчете **Карты пациентов** сделать группировку по полю *Код отделения*. Для каждой образованной группы строк выдать новый заголовок «Отделение Название отделения». Все строки, относящиеся к одному и тому же отделению, упорядочить по полю *Номер карты*. Сформированный отчет сохранить под именем **Карты пациентов по отделениям**.

5. В отчете **Карты пациентов по отделениям** предусмотреть для каждого отделения вычисление общего числа пациентов, находящихся на лечении на настоящий момент, и числа пациентов, выписавшихся в текущем месяце. Новому отчету дать имя **Карты пациентов по отделениям с итогами**.

Вариант 4. Отчеты к базе данных «Заработная плата сотрудников»

1. Для создания отчета использовать таблицу **Персонал**. В шапку отчета внести текст: «Список сотрудников». В отчет вывести поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Код должности*. Строки отчета упорядочить по полю *Табельный номер сотрудника*. Отчет сохранить под именем **Список сотрудников**.

2. В отчете **Список сотрудников** сделать группировку по полю *Номер отдела*. Для каждой образованной группы строк выдать новый заголовок «Отдел Номер отдела». Внутри отдела строки упорядочить по полю *Табельный номер сотрудника*. Созданный отчет сохранить под именем **Список сотрудников по отделам**.

3. Для создания отчета использовать таблицы **Персонал, Должности, Тарифная сетка**. В шапку отчета внести текст: «Личные счета сотрудников». В отчет вывести поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Пол, Номер отдела, Код должности, Должность, Номер разряда, Оклад*. Строки отчета упорядочить по полю *Табельный номер сотрудника*. Отчет сохранить под именем **Личные счета сотрудников**.

4. В отчете **Личные счета сотрудников** сделать группировку по полю *Код должности*. Для каждой образованной группы строк выдать новый заголовок «Должность Должность». Все строки, относящиеся к одному и тому же коду должности, упорядочить по полю *Табельный номер сотрудника*. Сформированный отчет сохранить под именем **Личные счета сотрудников по должностям**.

5. В отчете **Личные счета сотрудников по должностям** предусмотреть для всякой должности вычисление общего числа сотрудников и из них количества мужчин, женщин. Кроме того, необходимо для каждой должности найти средний оклад. Новому отчету дать имя **Личные счета сотрудников по должностям с итогами**.

Вариант 5. Отчеты к базе данных «Учет прихода товара»

1. Для создания отчета использовать таблицу **Товары**. В шапку отчета внести текст: «Список товаров». В отчет вывести поля: *Штрих-код, Наименование товара, Номер товарной группы, Единица измерения, Цена товара за единицу измерения*. Строки отчета упорядочить по полю *Штрих-код*. Отчет сохранить под именем **Список товаров**.

2. В отчете **Список товаров** сделать группировку по полю *Номер товарной группы*. Для каждой образованной группы строк выдать новый заголовок «Товарная группа Номер товарной группы». Внутри товарной группы строки упорядочить по полю *Штрих-код*. Созданный отчет сохранить под именем **Список товаров по группам**.

3. Для создания отчета использовать таблицы **Товары, Товарные группы, Приход**. В шапку отчета внести текст: «Учет прихода». В отчет вывести поля: *Дата прихода, Штрих-код, Наименование товара, Название группы, Единица измерения, Цена товара за единицу измерения, Количество, Поставщик*. Строки отчета упорядочить по полю *Штрих-код*. Отчет сохранить под именем **Учет прихода**.

4. В отчете **Учет прихода** сделать группировку по полю *Дата прихода*. Для каждой образованной группы строк выдать новый заголовок «Поступило на Дата прихода». Все строки с одной датой прихода упорядочить в порядке возрастания значений поля *Штрих-код*. Сформированный отчет сохранить под именем **Учет прихода по дате**.

5. В отчете **Учет прихода по дате** предусмотреть для всякой даты вычисление итоговой суммы прихода в денежном выражении. Новому отчету дать имя **Учет прихода по дате с итогом**.

Вариант 6. Отчеты к базе данных «Расписание движения поездов»

1. Для создания отчета использовать таблицу **Расписание**. В шапку отчета внести текст: «Расписание». В отчет вывести поля: *Номер поезда, Прибытие / Убытие, Дата прибытия (убытия) + Время прибытия (убытия)*. Строки отчета упорядочить по полю *Дата прибытия (убытия) + Время прибытия (убытия)*. Отчет сохранить под именем **Расписание**.

2. В отчете **Расписание** сделать группировку по полю *Прибытие / Убытие*. Для каждой образованной группы строк выдать новый заголовок «Характер поезда Прибытие/Убытие». Внутри группы строки упорядочить по полю *Номер поезда*. Созданный отчет сохранить под именем **Расписание по датам**.

3. Для создания отчета использовать таблицы **Расписание, Поезда, Пункты назначения**. Предварительно, если это не было сделано ранее, в таблицу **Расписание** добавить поле *Наполняемость* (тип числовой, одинарное с плавающей точкой, длина дробной части равна 2). Это поле отражает реальный процент занятости мест в поезде. Заполнить значения этого поля для всех записей таблицы **Расписание**. В шапку отчета внести текст: «Статистика». В отчет вывести поля: *Номер поезда, Проходящий / Местный, Прибытие / Убытие, Дата прибытия (убытия) + Время прибытия (убытия), Код пункта назначения, Название пункта назначения, Время следования, Расстояние, Нормативное количество посадочных мест, Наполняемость*. Строки отчета упорядочить по полю *Номер поезда*. Отчет сохранить под именем **Статистика**.

4. В отчете **Статистика** сделать группировку по полю *Код пункта назначения*. Для каждой образованной группы строк выдать новый заголовок «Пункт назначения Название пункта назначения». По каждому пункту назначения строки упорядочить в порядке возрастания значений поля *Дата прибытия (убытия) + Время прибытия (убытия)*. Сформированный отчет сохранить под именем **Статистика по пунктам назначения**.

5. В отчете **Статистика по пунктам назначения** предусмотреть для каждого пункта назначения и дате вычисление четырех итоговых сумм: общего количества посадочных мест в прибывающих (убывающих) поездах; фактическое число пассажиров приехавших (уехавших). Для вычисления фактического числа пассажиров приехавших (уехавших) необходимо использовать значения поля *Наполняемость*. Новому отчету дать имя **Статистика по пунктам назначения с итогами**.

Вариант 7. Отчеты к базе данных «Кадры»

1. Для создания отчета использовать таблицу **Сотрудники**. В шапку отчета внести текст: «Список сотрудников». В отчет вывести поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Пол, Дата рождения, Адрес*. Строки отчета упорядочить по полю *Табельный номер сотрудника*. Отчет сохранить под именем **Список сотрудников**.

2. В отчете **Список сотрудников** сделать группировку по полю *Пол*. Для каждой образованной группы строк выдать новый заголовок «Пол Пол». Внутри группы строки упорядочить по полю *Табельный номер сотрудника*. Созданный отчет сохранить под именем **Список сотрудников по полу**.

3. Для создания отчета использовать таблицы **Сотрудники, Отделы предприятия, Поступления и переводы**. Предварительно, если это не было сделано ранее, в таблицу **Поступления и переводы** внести и заполнить поле *Оклад*. В шапку отчета внести текст: «Передвижения сотрудников». В отчет вывести поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Пол, Дата поступления / перевода, Номер отдела, Название отдела, Фамилия руководителя, Оклад*. Строки отчета упорядочить по полю *Табельный номер сотрудника*. Отчет сохранить под именем **Передвижения сотрудников**.

4. В отчете **Передвижения сотрудников** сделать группировку по полю *Номер отдела*. Для каждой образованной группы строк выдать новый заголовок «Отдел Название отдела». Внутри отдела строки упорядочить по полю *Табельный номер сотрудника*. Сформированный отчет сохранить под именем **Передвижения сотрудников по отделам**.

5. В отчете **Передвижения сотрудников по отделам** предусмотреть для всякого отдела вычисление общего числа сотрудников и из них количества мужчин и женщин. Кроме того, необходимо для каждого отдела найти средний оклад и число сотрудников, поступивших (переведенных) в текущем году. Новому отчету дать имя **Передвижения сотрудников по отделам с итогами**.

Вариант 8. Отчеты к базе данных «Учебное расписание»

1. Для создания отчета использовать таблицу **Преподаватели**. В шапку отчета внести текст: «Список преподавателей». В отчет вывести поля: *Табельный номер преподавателя, Фамилия, Имя, Отчество, Телефон, Адрес, Название кафедры*. Строки отчета упорядочить по полю *Табельный номер преподавателя*. Отчет сохранить под именем **Список преподавателей**.

2. В отчете **Список преподавателей** сделать группировку по полю *Название кафедры*. Для каждой образованной группы строк выдать новый заголовок «Кафедра Название кафедры». Все строки, относящиеся к одной кафедре, упорядочить по полю *Табельный номер преподавателя*. Созданный отчет сохранить под именем **Список сотрудников по кафедрам**.

3. Для создания отчета использовать таблицы **Преподаватели, Учебные дисциплины, Расписание занятий**. В шапку отчета внести текст: «Расчет аудиторной нагрузки». В отчет вывести поля: *День недели, Лента, Аудитория, Название дисциплины, Фамилия, Имя, Отчество преподавателя, Название кафедры*. Строки отчета упорядочить по полям *День недели, Лента*. Отчет сохранить под именем **Расчет аудиторной нагрузки**.

4. В отчете **Расчет аудиторной нагрузки** сделать группировку по полю *Название кафедры*. Для каждой образованной группы строк выдать новый заголовок «Кафедра Название кафедры». Внутри кафедры строки упорядочить по полям *День недели, Лента*. Сформированный отчет сохранить под именем **Расчет аудиторной нагрузки по кафедрам**.

5. В отчете **Расчет аудиторной нагрузки по кафедрам** предусмотреть для всякой кафедры вычисление за семестр аудиторной нагрузки — суммарного числа часов проводимых занятий и планового количества часов по всем дисциплинам кафедры. Считать, что продолжительность семестра составляет 17 недель. Новому отчету дать имя **Расчет аудиторной нагрузки по кафедрам с итогами**.

Вариант 9. Отчеты к базе данных «Учет заказов покупателей»

1. Для создания отчета использовать таблицу **Товары**. В шапку отчета внести текст: «Список товаров». В отчет вывести поля: *Штрих-код, Наименование товара, Продуктовый / Промышленный, Номер партии, Срок годности*. Строки отчета упорядочить по полю *Штрих-код*. Отчет сохранить под именем **Список товаров**.

2. В отчете **Список товаров** сделать группировку по полю *Продуктовый/Промышленный*. Для каждой образованной группы строк выдать новый заголовок «Тип товара Продуктовый/Промышленный». Внутри группы строки

упорядочить по полям *Срок годности*, *Штрих-код*. Созданный отчет сохранить под именем **Список товаров по типам**.

3. Для создания отчета использовать таблицы **Товары**, **Покупатели**, **Заказы**. В шапку отчета внести текст: «Портфель заказов». Предварительно, если это не было сделано ранее, в таблицу **Товары** добавить и заполнить новое поле *Отпускная цена* (тип денежный, основной, длина дробной части равна 2). В отчете предусмотреть поля: *Дата заказа*, *Номер заказа*, *Штрих-код*, *Наименование товара*, *Отпускная цена*, *Количество*, *Название покупателя*, *Оптовый / Розничный*. Строки отчета упорядочить по полям *Дата заказа*, *Номер заказа*, *Штрих-код*. Отчет сохранить под именем **Портфель заказов**.

/ В отчете **Портфель заказов** сделать группировку по полю *Оптовый Розничный*. Для каждой образованной группы строк выдать новый заголовок «Тип покупателя Оптовый/Розничный». Для каждого типа покупателя строки упорядочить по полям *Дата заказа*, *Номер заказа*, *Штрих-код*. Сформированный отчет сохранить под именем **Портфель заказов по типам покупателей**.

5. В отчете **Портфель заказов по типам покупателей** предусмотреть для всякого типа покупателя вычисление общего количества заказов и суммарную стоимость заказов. Новому отчету дать имя **Портфель заказов по типам покупателей с итогами**.

Вариант 10. Отчеты к базе данных «Биржа труда»

1. Для создания отчета использовать таблицу **Вакансии**. В шапку отчета внести текст: «Список вакансий». В отчет вывести поля: *Код предприятия*, *Наименование предприятия*, *Сфера деятельности (отрасль)*, *Код профессии*, *Оклад*. Строки отчета упорядочить по полям *Код предприятия*, *Код профессии*. Отчет сохранить под именем **Список вакансий**.

2. В отчете **Список вакансий** сделать группировку по полю *Сфера деятельности (отрасль)*. Для каждой образованной группы строк выдать новый заголовок «Отрасль Сфера деятельности(отрасль)». Внутри группы строки упорядочить по полям *Код предприятия*, *Код профессии*. Созданный отчет сохранить под именем **Список вакансий по отраслям**.

3. Для создания отчета использовать таблицы **Профессии**, **Безработные**. В шапку отчета внести текст: «Список безработных». В отчет вывести поля: *Регистрационный номер безработного*, *Фамилия*, *Имя*, *Отчество*, *Адрес*, *Телефон*, *Образование*, *Код профессии*, *Название профессии*, *Стаж*. Строки отчета упорядочить по полю *Регистрационный номер безработного*. Отчет сохранить под именем **Список безработных**.

4. В отчете **Список безработных** сделать группировку по полю *Код профессии*. Для каждой образованной группы строк выдать новый заголовок «Профессия Название профессии». Внутри профессии строки упорядочить по полям *Регистрационный номер безработного*, *Стаж*. Сформированный отчет сохранить под именем **Список безработных по профессиям**.

5. В отчете **Список безработных по профессиям** предусмотреть для всякой профессии вычисление общего числа безработных, из них с высшим, незаконченным высшим, средним образованием. Кроме того, необходимо определить средний стаж безработных для каждой профессии. Новому отчету дать имя **Список безработных по профессиям с итогами**.

Контрольные вопросы

1. Что представляет собой отчет к базе данных? Опишите структуру и назначение отчёта.
2. Перечислите способы создания отчетов в СУБД Access. Дайте краткую характеристику каждому из перечисленных инструментов.
3. Приведите последовательность действий, необходимых для создания в мастере отчетов простейшего отчета на основе одной таблицы. Изменяется ли данная

- последовательность в ситуации усложнения отчетов: при использовании запроса, нескольких таблиц, нескольких запросов?
4. Какие разновидности мастера автоотчетов существуют? Чем они различаются между собой? Объясните, в чем заключается сходство и различие в работе мастера отчетов и мастера автоотчетов.
 5. Для чего предназначен режим предварительного просмотра? Опишите основные инструменты режима и их назначение. Продемонстрируйте, как в режиме предварительного просмотра производится печать отчета.
 6. Опишите назначение и свойства режима конструктора отчета. Перечислите основные окна и инструменты данного режима.
 7. Укажите, где в конструкторе располагается тело отчета. Дайте определение раздела отчета. Из каких разделов состоит тело отчета? Что может содержать каждый из перечисленных разделов?
 8. Что представляет собой элемент управления? Перечислите основные элементы управления отчетом и опишите их назначение.
 9. Продемонстрируйте работу с произвольным элементом управления: добавление элемента, изменение размера и местоположения элемента.
 10. Опишите назначение окна **Свойства** и его вкладок. Перечислите основные свойства отчета и элементов управления **Надпись**, **Поле**, **Флажок** и **Линия**.
 11. На примере элемента управления **Надпись** покажите, каким образом можно изменить цвет фона, цвет текста и оформление элемента.
 12. Что представляет собой источник данных для отчета? Какие объекты БД могут выступать в этой роли? Укажите способ задания источника данных в режиме конструктора.
 13. Как производится добавление в отчет сортировки данных, группировки данных? В чем заключается сходство и различие между ними?
 14. В каком случае в отчете появляются разделы **Заголовок группы** и **Примечание группы**? Что данные разделы могут содержать?
 15. Можно ли осуществлять группировку данных отчета не по всему значению поля, а его части?
 16. Какие параметры группы необходимо использовать для того, чтобы запретить разрыв группы и переносить ее на новую страницу при отображении отчета в режиме предварительного просмотра?
 17. Как осуществляется добавление в отчет вычисляемых полей и итоговых (статистических) функций?

Практическая работа №4. Простые вложенные запросы в SQL.

Целью практической работы является изучение представления запросов на языке SQL, синтаксисов и условий уточнения оператора запроса на выборку данных SELECT и операторов запросов на изменение (создание CREATETABLE, модификацию ALTERTABLE, удаление таблицы DROPTABLE, добавление данных INSERTINTO), структуры и особенности реализации вложенных запросов, операторов работы с запросами и их реализация на практике.

Порядок выполнения

Создать запрос в режиме SQL и выполнить его.

Практическая работа №11.

Вложенные запросы в SQL с применением операторов.

Произвести выборку данных, используя вложенный запрос (степень вложенности запросов и применение операторов IN, ANY, ALL, EX-ISTS зависят от задания).

Практическая работа №5

Вложенные запросы в SQL и оператор объединения.

1. Произвести выборку данных, используя вложенный запрос и оператор объединения UNION.
2. Ответить на один из контрольных вопросов к заданию.

Вариант 1. Вложенные запросы к базе данных «Студенческая научная работа»

1. Из таблицы **Преподаватели** выбрать преподавателей, работающих на кафедре прикладной математики.
2. Вывести список преподавателей со студентами, которые у них специализируются.
3. Вывести список преподавателей, у которых специализируются студенты группы М31. Для каждого преподавателя указать значения полей: *Фамилия, Имя, Отчество, Название кафедры*.
4. Найти список кафедр, на которых проходят специализацию студенты. Для каждой кафедры вывести *Название кафедры, Телефон*.
5. Определить студентов, специализирующихся на кафедре прикладной математики (она имеет код 07). Для каждого студента указать значения полей *Номер зачетной книжки, Фамилия, Имя, Отчество, Бюджетное / Платное обучение*.
6. Для кафедры прикладной математики сформировать общий список научных руководителей и специализирующихся у них студентов. Список должен содержать поля: *Фамилия, Имя, Отчество*.

Вариант 2. Вложенные запросы к базе данных «Абоненты телефонной компании»

1. Из таблицы **Телефоны** выбрать номера телефонов, абоненты которых являются пенсионерами.
2. Вывести общий список абонентов. Для каждого из них указать значения полей: *Фамилия, Имя, Отчество, Номер телефона, Адрес*.
3. Вывести список номеров телефонов, абоненты которых живут на определенной улице.
4. Найти список абонентов, которые еще не внесли плату за текущий месяц. Для каждого из них указать значения полей: *Фамилия, Имя, Отчество, Номер телефона, Адрес*.
и Определить перечень всех категорий абонентов, живущих на определенной улице. Предварительно, если это не было сделано ранее, в таблицу **Телефоны** необходимо добавить поле *Категория* (тип текстовый, длина 15) и заполнить его, используя такие виды категорий абонента: без льгот, пенсионер, ветеран, инвалид, многодетный и т.п.
6. Выявить всех должников – абонентов, которые еще не внесли плату за текущий месяц или эта плата не превысила 50 рублей. Для каждого абонента вывести значения полей *Регистрационный номер абонента, Фамилия, Имя, Отчество, Номер телефона, Адрес*.

Вариант 3. Вложенные запросы к базе данных «Статистика медицинского учреждения»

1. Выбрать из таблицы **Диагнозы заболевания**, для которых *Нормативное количество дней для стационарного лечения* составляет менее 10 дней.
2. Вывести список отделений с больными, находящимися в них на лечении. Для каждого пациента вывести значения полей: *Номер карты, Фамилия, Имя, Отчество, Ребенок / Взрослый, Дата поступления*.
3. Вывести список пациентов, для которых согласно диагнозу заболевания *Нормативное количество дней для стационарного лечения* превышает 7 дней. Для каждого

из них указать значения полей: *Номер карты, Фамилия, Имя, Отчество, Ребенок / Взрослый, Дата поступления.*

4 Вывести названия всех отделений, в которых еще находятся пациенты (с помощью полей *Дата поступления* и *Нормативное количество дней для стационарного лечения*).

5. Для отделения терапии (его код равен 205) определить список диагнозов пациентов, находящихся там на лечении. Список должен содержать поля: *Код диагноза, Диагноз.*

6. Перечислить пациентов отделения терапии, имеющих диагноз грипп (это заболевание имеет код ТО12) и всех пациентов отделения травматологии. Для каждого пациента указать значения полей: *Номер карты, Фамилия, Имя, Отчество, Ребенок / Взрослый, Дата поступления.*

Вариант 4. Вложенные запросы к базе данных «Заработная плата сотрудников»

1. Выбрать из таблицы Должности все должности с разрядом, превышающим 10.

2. Вывести список должностей с сотрудниками, работающими на них. Для каждого сотрудника указать значения полей: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Номер разряда.*

3. Вывести список сотрудников, имеющих *Номер разряда* больше 12. Для каждого сотрудника указать значения полей: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Должность, Номер разряда.*

4. Найти всех сотрудников, имеющих *Оклад* больше 4000 рублей. Для них вывести значения полей: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Должность, Номер разряда, Оклад.*

5. Для отдела с номером 10 определить перечень разрядов и соответствующих им окладов, которые имеют сотрудники данного отдела.

6. Сформировать список сотрудников, у которых *Номер разряда* больше 12 или *Оклад* больше 4000 рублей. Список должен содержать поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Номер отдела, Должность, Номер разряда, Оклад.*

Вариант 5. Вложенные запросы к базе данных «Учет прихода товара»

1. Из таблицы **Товары** выбрать товары, цена которых превышает 500 рублей. Для каждого товара выводить значения всех полей таблицы.

2. Вывести список товарных групп со всеми входящими в них товарами. Для каждого товара указать значения полей *Штрих-код, Наименование товара* и *Цена товара за единицу измерения.*

3. Найти наименования товарных групп, содержащих товары по цене больше 100 рублей.

4. Вывести список товаров, у которых *Количество* по приходу больше 20 единиц. Для каждого из них указать значения полей: *Штрих-код, Наименование товара, Цена товара за единицу измерения.*

5. Определить названия товарных групп, товары из которых были оприходованы в текущем месяце.

6. Для заданной товарной группы сформировать список товаров, у которых *Количество* по приходу больше 20 единиц или *Цена товара за единицу измерения* больше 100 рублей.

Вариант 6. Вложенные запросы к базе данных «Расписание движения поездов»

1. Вывести из таблицы **Расписание** информацию о поездах, прибывших сегодня (*Номер поезда, Код пункта назначения* и *Дата прибытия (убытия) + Время прибытия (убытия)*).

2. Вывести список пунктов назначений со всеми поездами, идущими в них. Использовать таблицы **Пункты назначений** и **Расписание**.

3. Вывести список пунктов назначений, из которых сегодня (в текущую дату) прибыли поезда.

4. Указать значения поля *Дата прибытия (убытия) + Время прибытия (убытия)* для всех местных поездов. Кроме того, для каждого поезда выдать *Номер поезда* и *Нормативное количество посадочных мест*.

5. Найти все пункты назначений, в которые идут местные поезда. Для каждого из них определить *Название пункта назначения, Время следования, Расстояние*.

6. Вывести значения поля *Дата прибытия (убытия) + Время прибытия (убытия)* для всех местных поездов или для тех поездов, у которых *Время следования* менее 24 часов. Дополнительно указать *Номер поезда, Код пункта назначения, Прибытие / Убытие*.

Вариант 7. Вложенные запросы к базе данных «Кадры»

1. Из таблицы **Сотрудники** выбрать всех молодых специалистов предприятия, т.е. сотрудников моложе 30 лет.

2. Вывести список названий отделов с работающими сотрудниками. Для каждого сотрудника указать поля: *Табельный номер сотрудника, Фамилия, Имя, Отчество, Дата поступления / перевода*.

3. Вывести список названий отделов, в которых работают сотрудники, которых *Дата поступления/перевода* относится к текущему году.

4. Вывести перечень значений поля *Дата поступления/перевода* для всех сотрудников, работающих в заданном отделе. Дополнительно показать поля *Табельный номер сотрудника* и *Оклад*.

5. Сформировать список названий отделов, в которых есть работающие сотрудники в возрасте старше 50 лет.

6. Найти всех сотрудников, работающих в заданном отделе или у которых *Дата поступления/перевода* относится к текущему году. Для каждого из них указать поля *Табельный номер сотрудника, Фамилия, Имя, Отчество, Дата рождения*.

Вариант 8. Вложенные запросы к базе данных «Учебное расписание»

1. Из таблицы **Преподаватели** выбрать преподавателей, работающих на кафедре прикладной математики.

2. Вывести расписание по дисциплинам, т.е. для каждой дисциплины указать поля: *Название дисциплины, День недели, Лента, Аудитория, Код преподавателя*.

3. Вывести для преподавателей кафедры прикладной математики время аудиторных занятий, т.е. значения полей: *День недели, Лента, Аудитория, Код дисциплины*.

4. В таблицу **Учебные Дисциплины** добавить и заполнить новое поле *Категория дисциплины* (тип текстовый, длина 15). Возможные значения это-го поля: альтернативная, обязательная, спецкурс. Найти расписание занятий по спецкурсам. В расписание включить *День недели, Лента, Аудитория, Фамилия, Имя, Отчество* преподавателя.

5. Определить названия всех альтернативных дисциплин кафедры прикладной математики.

6. Указать время аудиторных занятий для преподавателей кафедры прикладной математики, ведущих спецкурсы и время аудиторных занятий по любым альтернативным дисциплинам, т.е. значения полей: *День недели, Лента, Аудитория, Фамилия, Имя, Отчество* преподавателя.

Вариант 9. Вложенные запросы к базе данных «Учет заказов покупателей»

1. Из таблицы **Покупатели** выбрать всех оптовых покупателей.

2. Для каждого товара вывести портфель заказов, т.е. поля: *Штрих-Код, Наименование товара, Цена, Дата заказа, Номер заказа, Количество заказанного товара, Код покупателя*. Вывести список покупателей, сделавших заказы за текущий месяц. Для каждого из них указать значения полей: *Регистрационный номер покупателя, Название, Адрес*.

3. Вывести наименования товаров, заказанных оптовыми покупателями. Дополнительно указать поля: *Продуктовый/Промышленный* и *Срок годности*.

4. Сформировать портфель заказов для промышленных товаров. Портфель должен содержать поля: *Дата заказа, Номер заказа, Количество заказанного товара, Штрих-Код, Название покупателя, Адрес*.

5. Найти перечень заказов всех оптовых покупателей или заказов, включающих в себя промышленные товары. Этот перечень должен включать *Дата заказа, Номер заказа, Количество заказанного товара, Штрих-код*.

Вариант 10. Вложенные запросы к базе данных «Биржа труда»

1. Из таблицы Вакансия вывести всю информацию по вакансиям частных предприятий.

2. Вывести список названий профессий с безработными по ним. Для безработных следует указать значения полей: *Регистрационный номер безработного, Фамилия, Имя, Отчество, Номер телефона, Адрес, Стаж*.

3. Вывести список названий профессий безработных с высшим образованием.

4. Найти всех безработных, имеющих профессии, по которым в вакансиях предлагается оклад выше 5000 рублей. Для каждого из них следует указать значения полей: *Регистрационный номер безработного, Фамилия, Имя, Отчество, Номер телефона, Адрес, Название профессии, Стаж*.

5. Сформировать перечень названий профессий, вакансии по которым предлагают частные предприятия.

6. Определить список названий профессий безработных с высшим образованием или профессий, по которым в вакансиях частных предприятий предлагается оклад выше 5000 рублей.

Контрольные вопросы

1. Какой режим работы с запросами в СУБД Access предназначен для создания и выполнения команд языка SQL?
2. Какая разновидность запроса создается в результате выполнения оператора SELECT? Опишите синтаксис данного оператора. Приведите пример тривиального запроса.
3. Для чего в операторе SELECT используются ключевые слова ALL и DISTINCT?
4. С помощью какого дополнительного оператора в запросе на выборку задается условие отбора записей? Какие предикаты и логические операторы могут использоваться в условии отбора? Продемонстрируйте использование условия отбора записей в любом запросе.
5. Что позволяет проверять в условии отбора предикат BETWEEN? Приведите пример условия отбора записей с предикатом BETWEEN.
6. Расшифруйте назначение предиката LIKE в условии отбора записей. Продемонстрируйте использование данного предиката в условии отбора записей.
7. Для чего в условии отбора записей используется предикат IN? Приведите пример запроса с данным предикатом.
8. Укажите дополнительный оператор в запросе на выборку, который позволяет организовать сортировку. Как можно сменить порядок сортировки? Отсортируйте данные любого запроса.
9. Какой дополнительный оператор, используемый в условии отбора, осуществляет группировку данных запроса? Приведите пример запроса с группировкой данных.
10. Как в операторе SELECT создаются вычисляемые поля? В каких частях запроса они могут находиться? Приведите примеры.
11. Перечислите операции объединения таблиц, которые существуют в СУБД Access. Объясните различия между ними.

12. Что представляет собой запрос на изменение данных? Перечислите основные типы запросов на изменение и опишите синтаксис соответствующих им команд на языке SQL.
13. Опишите назначение, структуру и особенности реализации вложенных запросов. Приведите пример вложенного запроса.
14. Какая степень вложенности запросов допустима в Access? Опишите порядок выполнения запросов из сложного запроса произвольной вложенности.
15. Для чего в запросах, содержащих вложенных запрос, используется предикат IN? Укажите, в чем заключается специфика использования с вложенными запросами предикатов ALL, ANY, EXISTS. Приведите примеры.
16. Продемонстрируйте, как осуществляется связь между внешним и вложенным запросом без использования каких-либо предикатов.
17. Опишите назначение и структуру операторов работы с запросами: объединения, пересечения и вычитания запросов. Как перечисленные операторы реализуются в СУБД Access?

Практическая работа №5. Простейшие формы

Цель: изучение назначения и свойств форм и управляющих элементов формы (в т.ч. полей, подписей, переключателей, списков, команд-ных кнопок и т.д.), использование командных кнопок для передвижения по записям таблицы, обработки записей, поиска записей, работы с запросом, от-чётом, приложением и формой, создание главной формы и подчиненной к ней, а также практика создания форм различной сложности с помощью режимов конструктора и мастера форм.

Порядок выполнения

Построить три простейшие формы ввода и редактирования данных трех соответствующих таблиц базы данных. Строки (записи) таблиц предварительно упорядочить. Создать кнопки для передвижения по записям таблиц: **Вперед, Назад, Первая запись, Последняя запись, Добавить новую запись**. Организовать поиск записей по условию.

Практическая работа №6. Форма ввода данных.

Построить форму ввода для работы с данными нескольких таблиц, используя выбор из списка.

Практическая работа №7. Администрирование баз данных.

Целью данной практической работы является изучение различных средств защиты и контроля целостности данных, а также приемов управления доступом к БД и ее объектам и их реализация на практике. Задание рассчитано на 4 часа аудиторных занятий.

Порядок выполнения

1. Изучение средств защиты данных – различных приемов управления доступом к базе данных и ее объектам в MSAccess.

Практическая работа №8. Создание базы данных в среде MicrosoftSQLServer

В составе Microsoft Visual Studio 2008 находится сервер баз данных Microsoft SQL Server 2005 Express Edition. От полнофункционального сервера данных он отличается только ограничением размера базы данных в 2 гигабайта, что позволяет производить разработку и тестирование приложений баз данных.

Для работы по созданию базы данных и таблиц будем использовать Microsoft SQL Server Management Studio Express. Данный программный продукт является свободным распространяемым и доступен для скачивания в Интернет.

Определение структуры базы данных

Внешний вид окна программы Microsoft SQL Server Management Studio Express приведен на рис. 1.

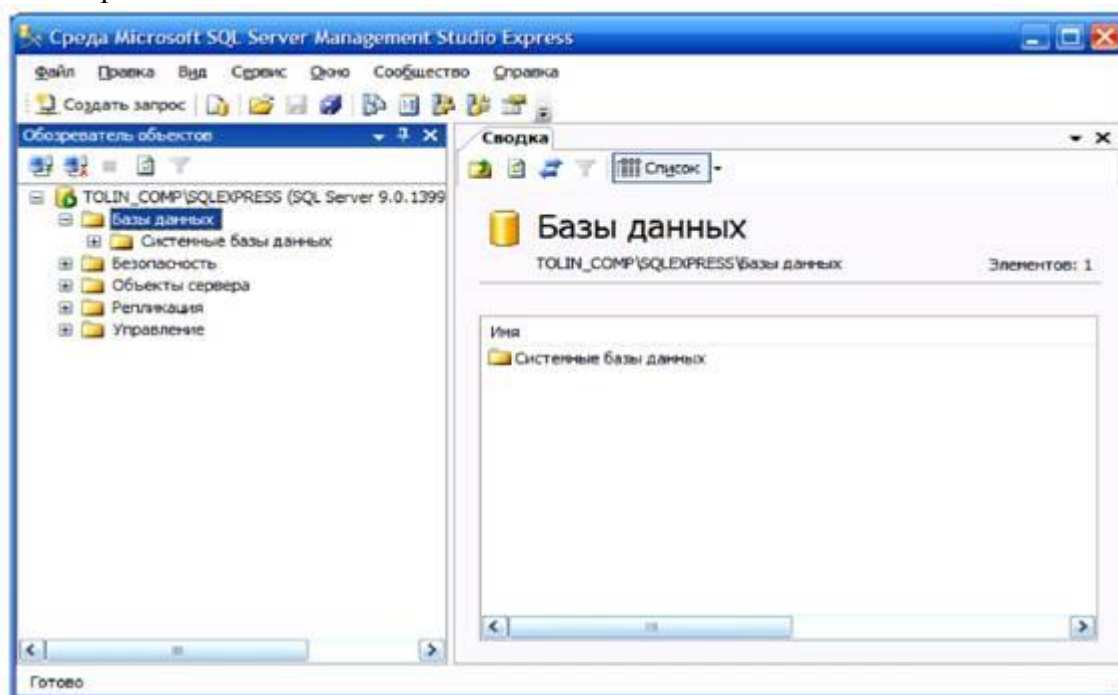


Рис. 1. Внешний вид окна программы Microsoft SQL Server Management Studio Express

Для создания базы данных необходимо кликнуть правой кнопкой мыши на пункте «Базы данных» и выбрать пункт меню «Создать базу данных». Окно создания БД представлено на рис. 2.

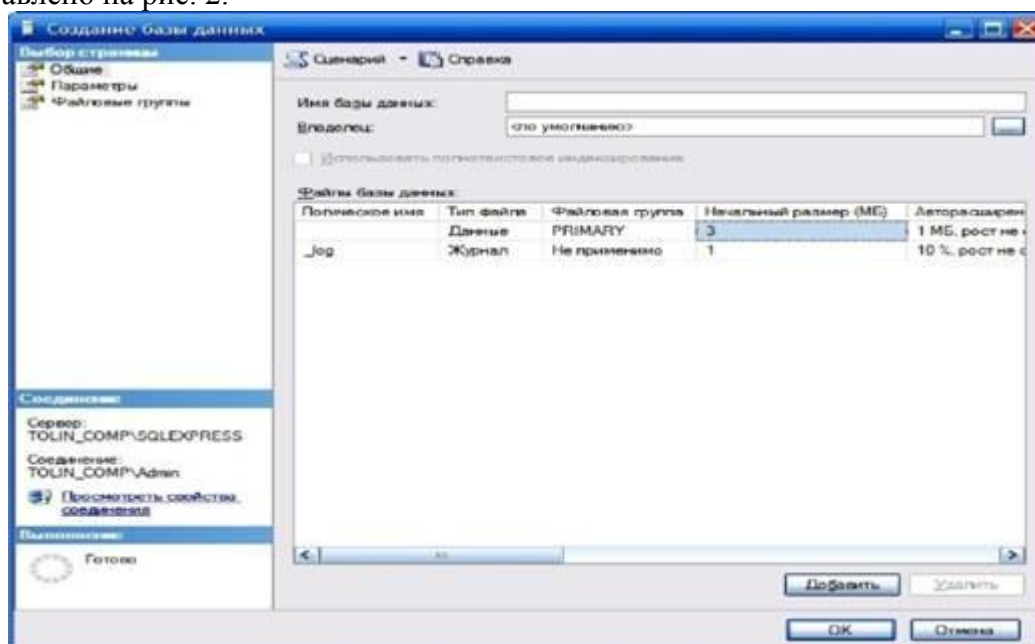


Рис. 2. Окно создания БД

В данном окне задается имя базы данных, имена и пути к файлам базы данных, начальный размер файлов и шаг увеличения размера БД в случае необходимости. После нажатия кнопки «ОК» созданная БД появляется в списке баз данных (рис. 3).

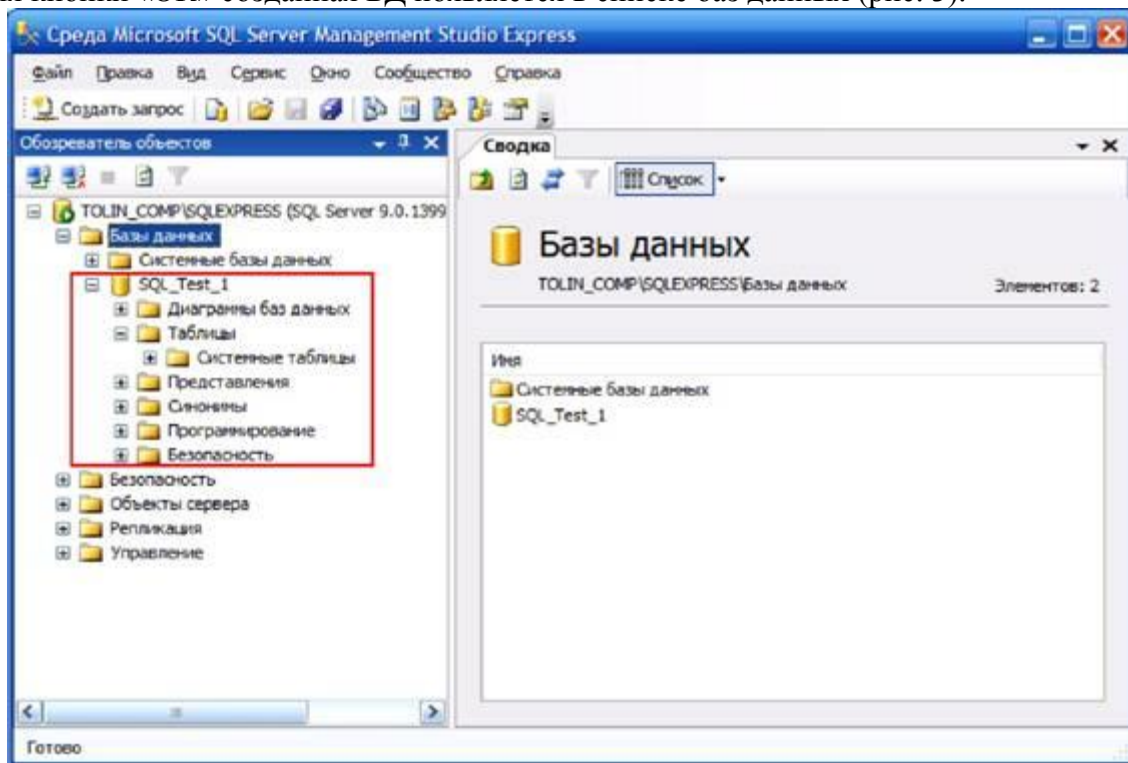


Рис. 3. Вид Management Studio с созданной базой данных

Практическая работа №9. Создание связей базы данных в среде MicrosoftSQLServer.

Создание связей происходит путем совмещения связываемых полей. Результатом становится появление окна создания отношения (рис. 7).

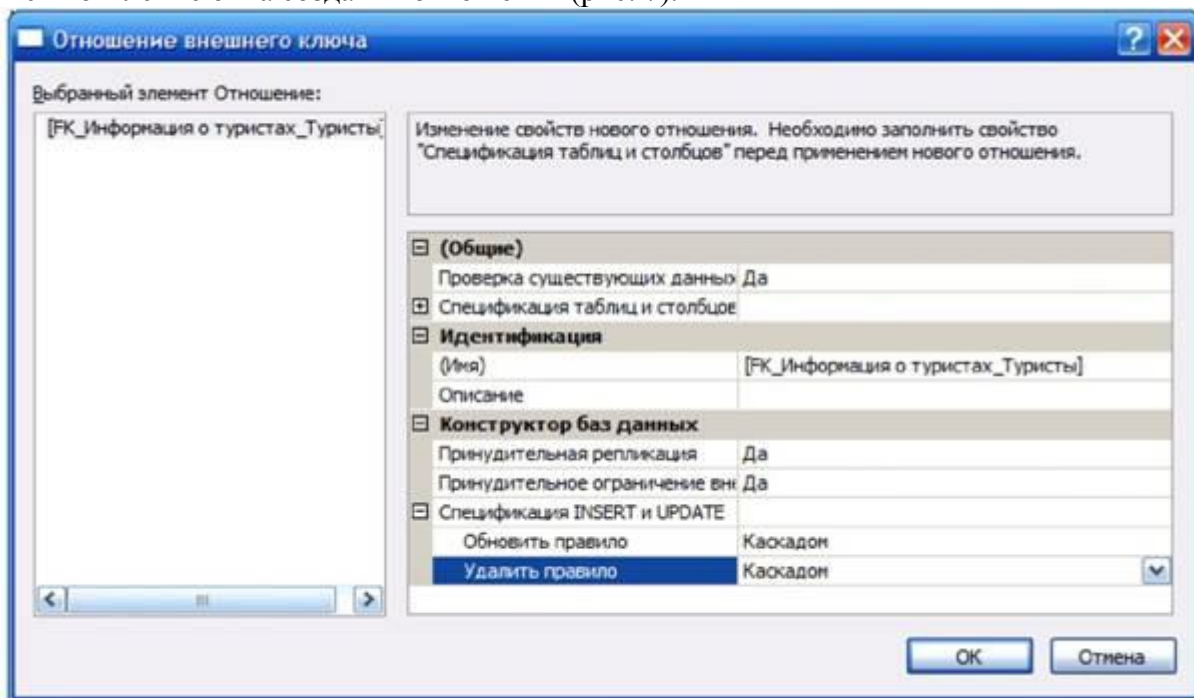


Рис. 7. Создание отношения между таблицами БД

Особо отметим пункт «Спецификация INSERT и UPDATE», задающий правила обновления и удаления связанных данных в таблицах.

После создания остальных таблиц и их связей схема данных будет выглядеть следующим образом (рис. 8).

В отличие от схемы данных Microsoft Access, здесь линии, отображающие связи по умолчанию, не привязываются графически к первичным и вторичным полям. Однако при щелчке левой кнопкой на любой связи в панели свойств появляется информация о выбранном отношении.

Завершив работу со схемой данных, сохраняем ее. Отметим, что в SQL Management Studio, в отличие от Access, для одной базы данных может быть создано несколько диаграмм (рис. 9).

Данная возможность является полезной для баз данных с очень большим количеством таблиц, так как одна общая диаграмма была бы слишком нагруженной.

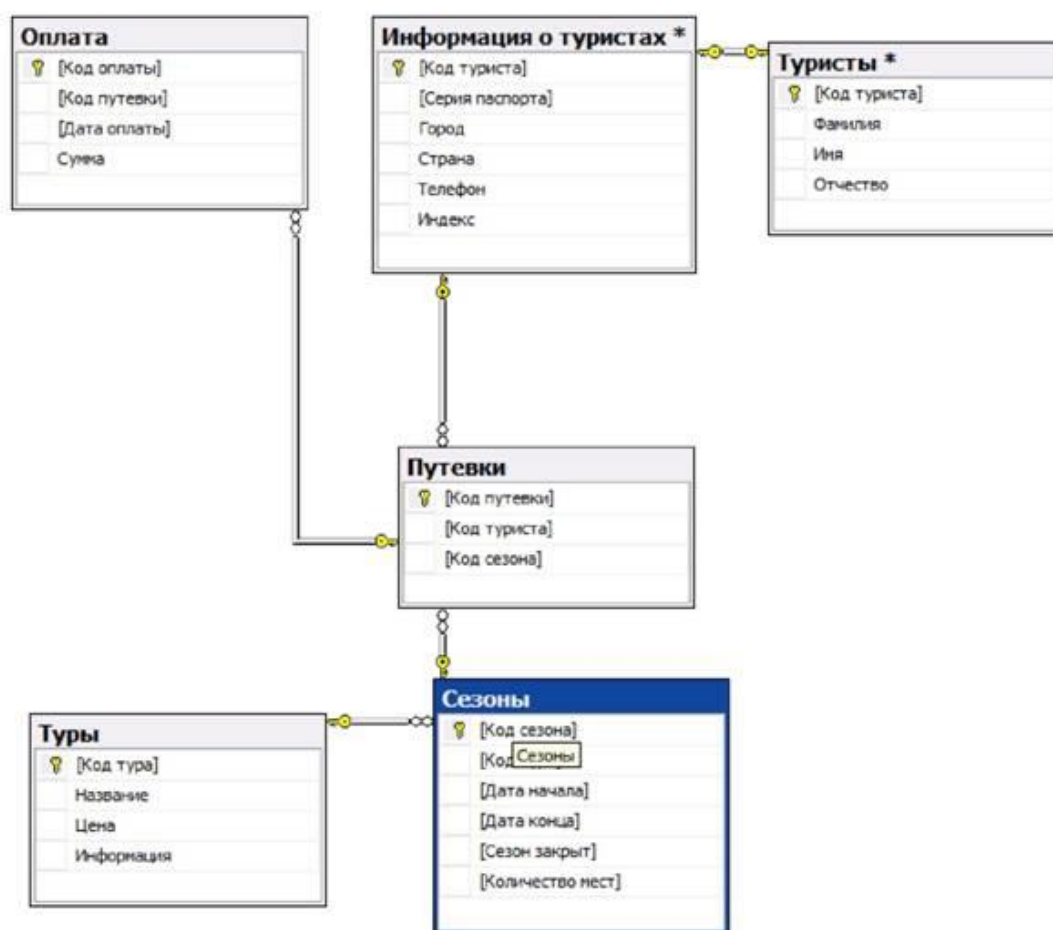


Рис. 8. Схема базы данных BDTur_firmSQL

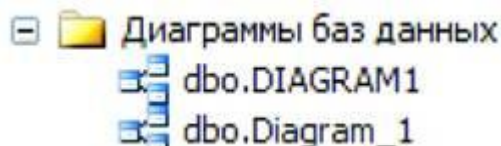


Рис. 9. Несколько диаграмм для одной БД

Контрольные вопросы и задания.

1. Является ли схемой реляционной БД следующий набор из пяти абстрактных атрибутов: А В С А М? Почему?
2. Является ли отношением следующий набор записей со схемой [ABC]: (а, в, с), (а1, в, с), (а1, в1, с), (а2, в, с1), (а, в, с), (а1, в, с1), (а2, в1, с)? Почему?
3. Пусть дана следующая схема отношения [СЛУЖАЩИЙ, РУКОВОДИТЕЛЬ,

5. Выделите ключи для следующей схемы отношения [№2 РЕЙСА, АЭРОПОРТ НАЗНАЧЕНИЯ, ВЫХОД, ДАТА, ВРЕМЯ], которая означает, что посадка на рейс № РЕЙСА, вылетающий в город АЭРОПОРТ НАЗНАЧЕНИЯ, осуществляется через выход номер ВЫХОД, дата отправления рейса - ДАТА, время вылета - ВРЕМЯ.

7. Обязательно ли пересечение двух наборов атрибутов, содержащих ключи отношения, является ключом?

9. Постройте словарь данных для отношения из заданий 3,4.

12. Расширьте возможности приложения-примера БД менеджера турфирмы, изменив разработанную схему БД таким образом, чтобы можно было хранить несколько контактных телефонов для каждого туриста.

Перенос файла БД MicrosoftSQL на другой компьютер

The screenshot shows the Microsoft SQL Server Enterprise Manager interface. In the left-hand pane, the 'Обозреватель объектов' (Object Explorer) displays the server 'TOLIN_COMP\SQLEXPRESS (SQL Server 9.0.1399)'. Under the 'Базы данных' (Databases) folder, the 'BDTur_firmSQL' database is selected. A right-click context menu is open, showing options such as 'Создать базу данных...', 'Создать запрос', 'Создать сценарий для базы данных', 'Задачи', 'Переименовать', 'Удалить', 'Обновить', and 'Свойства'. The 'Задачи' (Tasks) option is expanded, and the 'Отсоединить...' (Detach...) option is highlighted with a red rectangle. The main pane on the right shows the 'Сводка' (Summary) tab for the 'BDTur_firmSQL' database, indicating its location as 'TOLIN_COMP\SQLEXPRESS\Базы данных\BDTur_firmSQL'.

Рис. 10. Отсоединение выбранной базы данных от сервера

Появляется диалоговое окно «Отсоединение базы данных». Подтверждаем отсоединение, нажимая кнопку «ОК», - и база отсоединена. Теперь нужные файлы доступны для копирования.

Для присоединения базы данных на другом компьютере запускаем SQL Management Studio, выделяем ветку «Базы данных» и в контекстном меню выбираем «Присоединить» (рис. 11).

В появившемся окне указываем расположение файла базы данных BDTur_firmSQL.mdf - файл отчетов присоединится автоматически - и нажимаем «ОК». Присоединившаяся база данных немедленно отображается в папке «Базы данных». Следует отметить, что после присоединения БД может потребоваться настройка пользователей БД и прав доступа.

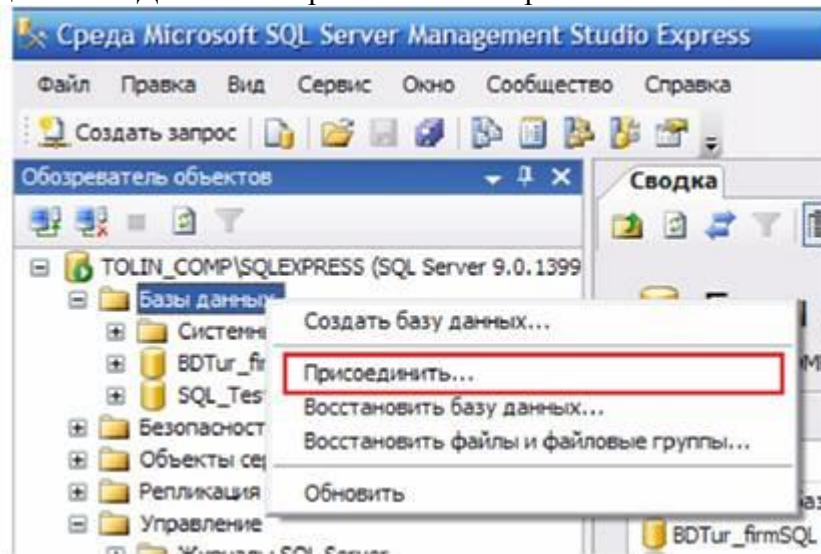


Рис. 11. Присоединение базы данных

Практическая работа №11. Команды изменения данных языка DML.

Значения могут быть помещены и удалены из полей тремя командами языка DML (Язык Манипулирования Данными):

- insert (вставить),
- update (изменить),
- delete (удалить).

Команда insert имеет свои особенности.

- При указании значений конкретных полей вместо использования каких-либо значений можно применить ключевое слово DEFAULT.

- Вставка пустой строки приводит к добавлению пробела, а не значения NULL.

- Строки и даты задаются в апострофах.

- Можно задавать NULL явно, а можно задавать DEFAULT.

Например:

```
insert into ClientInfo (FirstName, LastName, Address, Phone) values('Petr','Petrov','Chehova 13','1234567');
```

Однократное выполнение этого запроса (нажатие клавиши F5 один раз) приводит к добавлению одной записи. Добавим еще несколько записей, изменяя значения values:

```
insert into ClientInfo (FirstName, LastName, Address, Phone) values('Ivan','Ivanov','Naberejnaya 13,,,'1234568'); insert into ClientInfo (FirstName, LastName, Address, Phone) values(null,'Sidorov','Naberejnaya 25','1234569');
```

Извлечем все записи созданной таблицы (рис. 19):

```
select * from ClientInfo;
```

	FirstName	LastName	Address	Phone
1	Petr	Petrov	Chehova 13	1234567
2	Ivan	Ivanov	Naberejnaya 13	1234568
3	NULL	Sidorov	Naberejnaya 25	1234569

Рис. 19. Все записи таблицы ClientInfo

Отметим, что третья строка содержит значение null, а не текстовую строку «null».

Команда update позволяет изменять заданные значения записей:

update ClientInfo set FirstName = 'Andrey' where FirstName = 'Petr';

В этом случае в первой записи поля FirstName значение Petr изменится на Andrey (рис. 20).

	FirstName	LastName	Address	Phone
1	Andrey	Petrov	Chehova 13	1234567
2	Ivan	Ivanov	Naberejnaya 13	1234568
3	NULL	Sidorov	Naberejnaya 25	1234569

Рис. 20. Изменение одной записи

Отметим, что если не указывать условие, определяющее значение, которое необходимо изменить, команда update затронет все записи.

Команда delete удаляет записи из таблицы.

delete from ClientInfo where LastName like 'Petrov';

Результатом этого запроса будет удаление первой записи из таблицы ClientInfo.

Если не задавать условие, определяющее данные, которые необходимо удалить, то будут удалены все данные таблицы.

Запросы с командами insert, update и delete могут содержать в себе все прочие конструкции языка SQL.

Контрольные вопросы и задания.

1. Напишите SQL-запросы для вывода на экран содержания всех таблиц БД (для каждой таблицы свой запрос, см. пример из п. 2.1. select * from Туристы;).
2. Добавьте к соответствующим запросам задания п. 1 сортировку по фамилиям и именам.
3. Что будет выведено на экран в результате выполнения следующего запроса: select Фамилия, Имя, Отчество from Туристы order by Имя desc, Отчество asc; ?
4. Напишите SQL-запрос, который позволит отобразить всех туристов, проживающих в заданном городе. Используйте сортировку при выводе.
5. Посчитайте туристов с одинаковыми именами.
6. А как посчитать туристов с одинаковыми отчествами? Ведь слова «Иванович» и «Ивановна» одинаковые отчества, но не одинаковые строки, которые хранятся в базе данных.
7. Как определить среднюю цену по турам в Париж, (например, цена может меняться в зависимости от сезона)?
8. Как будет выглядеть таблица «Туристы» после выполнения следующей SQL-команды: update Туристы set Имя = 'Владимир'?
9. 10. Что произойдет с таблицей «Туры» после выполнения SQL-команды: delete from Туры?
11. Выясните с помощью SQL-запроса к БД, кто из туристов еще не оплатил свои путевки? Подсчитайте их количество и общую сумму недоплаты.
12. Распечатайте все предлагаемые турфирмой туры в алфавитном порядке.

13. Составьте с помощью оператора update SQL-команду для переименования города Ульяновска в Симбирск в информации о туристах.
14. Распечатайте все предлагаемые турфирмой туры с сезонами.
15. Выведите полную информацию о туристах, выкупивших путевки на какой-нибудь определенный тур и сезон.
16. С учетом внесенных изменений в структуру БД по заданию п.12 раздела 1.5. постройте SQL-запрос, выводящий полную контактную информацию о туристах, имеющих долги по оплате своих путевок.
17. С помощью операторов добавления в БД перенесите соответствующую информацию из отношений «Туристы» и «Информация о туристах» в отношение ClientInfo.

Практическая работа №12. Простые запросы к БД.

Перейдем к созданию своих собственных процедур. Создадим новый бланк запросов и введем следующий запрос:

create procedure proc1 **asselect** [Код туриста], Фамилия, Имя, Отчество **from** Туристы

Здесь createprocedure - оператор, указывающий на создание хранимой процедуры, proc1 - ее название, далее после оператора as следует обычный SQL-запрос. Квадратные скобки необходимы для указания поля таблицы, в названии которого содержится пробел. После выполнения запроса появится сообщение:

Выполнение команд успешно завершено.

TheCOMMAND(s) completedsuccessfully.

Данное сообщение означает, что все сделано правильно и команда создала процедуру proc1. Убедиться в этом можно, развернув ветку «Программирование - Хранимые процедуры» в среде ManagementExpress (рис. 21).

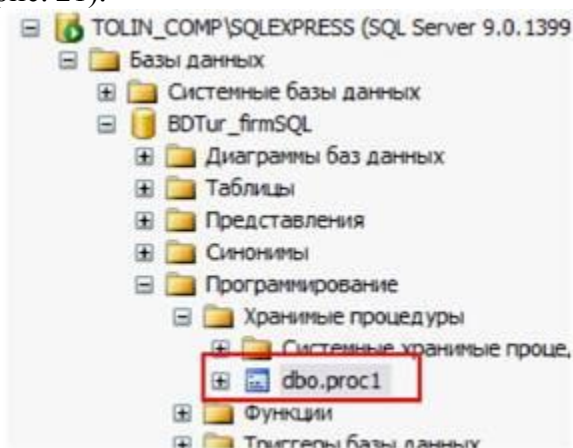


Рис. 21. Созданная хранимая процедура

Для просмотра результата вызываем ее:

execproc1

Появляется уже знакомое извлечение всех записей таблицы «Туристы» со всеми записями (рис. 22).

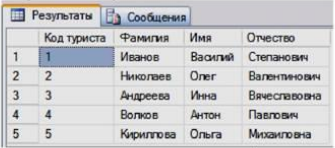
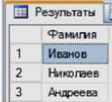
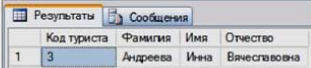
Из полученного результата видно, что создание содержимого хранимой процедуры не отличается ничем от создания обычного SQL-запроса.

В таблице 2 приведены примеры хранимых процедур.

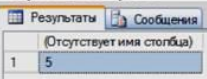
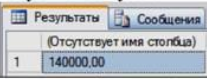
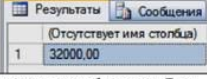
Результаты		Сообщения		
	Код туриста	Фамилия	Имя	Отчество
1	1	Иванов	Василий	Степанович
2	2	Николаев	Олег	Валентинович
3	3	Андреева	Инна	Вячеславовна
4	4	Волков	Антон	Павлович
5	5	Кириллова	Ольга	Михайловна

Рис. 22. Результат запуска процедуры proc1

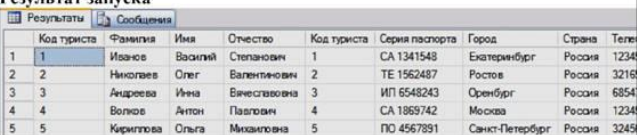
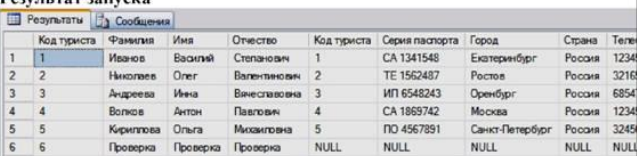
Таблица 2. Примеры хранимых процедур

№	SQL-конструкция для создания	Команда для извлечения	Описание
1	create procedure proc1 as select [Код туриста], Фамилия, Имя, Отчество from Туристы	exec proc1	Вывод всех записей таблицы «Туристы»
Результат запуска 			
2	create procedure proc2 as select top 3 Фамилия from туристы	exec proc2	Вывод первых трех значений поля «Фамилия» таблицы «Туристы»
Результат запуска 			
3	create procedure proc3 as select * from туристы where Фамилия = 'Андреева'	exec proc3	Вывод всех полей таблицы «Туристы», содержащих в поле «Фамилия» значение «Андреева»
Результат запуска 			

Продолжение таблицы 2

№	SQL-конструкция для создания	Команда для извлечения	Описание
4	create procedure proc4 as select count (*) from Туристы	exec proc4	Подсчет числа записей таблицы «Туристы»
Результат запуска 			
5	create procedure proc5 as select sum(Сумма) from Оплата	exec proc5	Подсчет значений поля Сумма таблицы Оплата
Результат запуска 			
6	create procedure proc6 as select max(Цена) from Туры	exec proc6	Вывод максимального значения поля Цена таблицы Туры
Результат запуска 			
7	create procedure proc7 as select min(Цена) from Туры	exec proc7	Вывод минимального значения поля Цена таблицы Туры
Результат запуска 			
8	create procedure proc8 as select * from Туристы where Фамилия like '%и%'	exec proc8	Вывод всех записей таблицы Туристы, содержащих в значении поля Фамилия букву «и» (в любой части слова)
Результат запуска 			

Окончание таблицы 2

№	SQL-конструкция для создания	Команда для извлечения	Описание
9	create procedure proc9 as select * from Туристы inner join [Информация о туристах] on Туристы.[Код Туриста]= [Информация о туристах].[Код Туриста]	exec proc9	Соединение двух таблиц «Туристы» и «Информация о туристах» по полю «Код туриста» и вывод полной связанной информации из обеих таблиц
Результат запуска 			
10	create procedure proc10 as select * from Туристы left join [Информация о туристах] on Туристы.[Код Туриста]= [Информация о туристах].[Код Туриста]	exec proc10	Прежде чем создать эту процедуру и затем ее выполнить, добавим в таблицу «Туристы» базы данных BDTur_firm новую строку с произвольными значениями. В результате в таблице «Туристы» у нас получится 6 записей, а в связанной с ней таблице «Информация о туристах» – 5. Создаем хранимую процедуру и запускаем ее
Результат запуска 			

Операция innerjoin объединяет записи из двух таблиц, если поле (поля), по которому связаны эти таблицы, содержат одинаковые значения. Общий синтаксис выглядит следующим образом:

from таблица1**innerjoin** таблица2 **on** таблица1.поле1< оператор сравнения>
таблица2.поле2

Операция **leftjoin** используется для создания так называемого левого внешнего соединения. С помощью этой операции выбираются все записи первой (левой) таблицы, даже если они не соответствуют записям во второй (правой) таблице. Общий синтаксис имеет вид:

from таблица1**leftjoin** таблица2 **on** таблица1.поле1< оператор сравнения>
таблица2.поле2.

Основная литература:

1. Нестеров, С.А. Интеллектуальный анализ данных средствами MS SQLServer 2008 / С.А. Нестеров. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 338 с. : ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429083>
2. Сирант, О.В. Работа с базами данных / О.В. Сирант, Т.А. Коваленко. - 2-е изд., испр. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 150 с. : схем., ил. - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=428978>
3. Кумскова, И. А. Базы данных: учебник для СПО / И. А. Кумскова. - М.: КНОРУС, 2016. - 488 с.
4. Лазицкас, Е.А. Базы данных и системы управления базами данных : учебное пособие / Е.А. Лазицкас, И.Н. Загумённикова, П.Г. Гилевский. - Минск : РИПО, 2016. - 267 с. : ил. - Библиогр. в кн. - ISBN 978-985-503-558-0; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=463305>

Дополнительная литература:

1. Кузнецов, С. Введение в модель данных SQL : курс / С. Кузнецов. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 351 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. - ISBN 5-9556-00028-0; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429087>
2. Кузнецов, С. Введение в реляционные базы данных / С. Кузнецов. - 2-е изд., исправ. - Москва : Национальный Открытый Университет «ИНТУИТ», 2016. - 248 с. : ил. - (Основы информационных технологий). - Библиогр. в кн. ; То же [Электронный ресурс]. - URL: <http://biblioclub.ru/index.php?page=book&id=429088>
3. Швецов В.И. Базы данных [Электронный ресурс]/ Швецов В.И.— Электрон. текстовые данные.— М.: Интернет-Университет Информационных Технологий (ИНТУИТ), 2016.— 218 с.— Режим доступа: <http://www.iprbookshop.ru/52139.html>.— ЭБС «IPRbooks»

