


Министерство просвещения Республики Башкортостан
Государственное бюджетное профессиональное образовательное учреждение
Октябрьский многопрофильный профессиональный колледж

Утверждено
На заседании МС
профессионального цикла протокол № 1 от «30» августа 2024 г.
протокол № 1 от «30» августа 2024 г.

Рассмотрено
на заседании ПЦК
Председатель ПЦКГ.  Ф.Ямаева

Методические рекомендации
по обеспечению практической работы
по учебной дисциплине ОП.03 Базы данных

Разработал преподаватель
Фонакова Н.П.

2024 г.

Пояснительная записка

Методические указания для выполнения практических работ составлены в соответствии Рабочей программой учебной дисциплины ОП.03 «Базы данных» и предназначены для выполнения практических работ студентами очного отделения по специальности 09.01.03 Оператор информационных систем и ресурсов.

В процесс освоения дисциплины у студентов должны формироваться общие компетенции (ОК):

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, предпринимательскую деятельность в профессиональной сфере, использовать знания по правовой и финансовой грамотности в различных жизненных ситуациях;
ОК 04. Эффективно взаимодействовать и работать в коллективе и команде;
ОК 05. Осуществлять устную и письменную коммуникацию на государственном языке Российской Федерации с учетом особенностей социального и культурного контекста;
ОК 06. Проявлять гражданско-патриотическую позицию, демонстрировать осознанное поведение на основе традиционных российских духовно-нравственных ценностей, в том числе с учетом гармонизации межнациональных и межрелигиозных отношений, применять стандарты антикоррупционного поведения;
ОК 07. Содействовать сохранению окружающей среды, ресурсосбережению, применять знания об изменении климата, принципы бережливого производства, эффективно действовать в чрезвычайных ситуациях;
ОК 08. Использовать средства физической культуры для сохранения и укрепления здоровья в процессе профессиональной деятельности и поддержания необходимого уровня физической подготовленности;
ОК 09. Пользоваться профессиональной документацией на государственном и иностранном языках.

В процесс освоения дисциплины у студентов должны формироваться профессиональные компетенции (ПК):

ПК 1.1. Выполнять ввод и обработку текстовых данных.
ПК 1.2. Выполнять преобразование данных, связанных с изменениями структуры документов.
ПК 1.3. Выполнять разметку и форматирование документов различных форматов.
ПК 1.4. Конвертировать аналоговые данные в цифровые.
ПК 1.5. Выполнять подготовку цифровых данных для дальнейшей обработки и архивирования.
ПК 1.6. Формировать запросы для получения информации в базах данных.
ПК 1.7. Выполнять операции с объектами базы данных.
ПК 2.1. Структурировать цифровые данные для публикации.
ПК 2.2. Размещать и обновлять информационный материал через систему управления контентом.
ПК 2.3. Устанавливать и разграничивать права доступа к разделам веб-ресурса.
ПК 2.4. Собирать статистику по результатам работы веб-ресурса.

- Развивать навык проектирования баз данных (ER-моделирование, нормализация).
- Закреплять практические навыки работы с СУБД: создание схем, заполнение данными, выполнение запросов, настройка индексов, обеспечения целостности данных.
- Овладеть методиками анализа требований, документирования архитектуры БД и результатов.
- Формировать дисциплину учёта версий, тестирования запросов и обеспечение безопасности.
- Развивать навыки работы с инструментами разработки БД и базовыми концепциями OLTP/OLAP.

2) Виды и формы самостоятельной работы

- Теоретические задания: принципы моделирования баз данных, нормализация, целостность данных, транзакции ACID, индексация.
- Практические задания в СУБД: создание схемы, создание таблиц, ограничений, представлений, триггеров, хранимых процедур, транзакций, резервное копирование.
- Лабораторные работы: настройка среды, загрузка тестовых данных, выполнение комплексных SQL-запросов (JOIN, вложенные запросы, группировки, подзапросы), оптимизация.
- Мини-проекты: разработка небольшой базы данных для конкретной предметной области (магазин, библиотека, система резерваций) с полным циклом — ERD, DDL/DML, примеры запросов и отчётов.
- Аналитические задания и обзоры: сравнение СУБД (PostgreSQL, MySQL, SQL Server, Oracle, NoSQL), обзор технологий хранения данных, выбор технологий под задачу.
- Контроль знаний: задания на SQL, тесты по архитектуре БД, задачи на логику.
- Рефлексия и самооценка: краткий разбор сделанного и план по улучшению.

3) Требования к оформлению и представлению материалов

- Структура отчета (пример):
- Титульный лист
- Аннотация
- Цели и задачи
- Описание методики выполнения
- Ход выполнения (пошагово)
- Результаты (таблицы, диаграммы, примеры запросов)
- Выводы
- Приложения (DDL/DML-скрипты, скриншоты, диаграммы)
- Источники
- Формат текста: ясный стиль, корректный язык, единый стиль оформления.
- Иллюстрации: подписывать схемы, диаграммы, скриншоты; приводить источники данных.
- Программы и коды: сопровождать пояснениями; обеспечить работоспособность скриптов.
- Электронная подача: PDF или установленный кафедрой формат; наименование файла по установленному шаблону.
- Оформление ссылок: единый стиль (IEEE/APA по договоренности); минимизация копипаста.

4) Алгоритм выполнения самостоятельной работы (практикум)

- Шаг 1. Изучение задания и формулировка целей.
- Шаг 2. Планирование: сроки, ресурсы, этапы реализации.
- Шаг 3. Проектирование: сбор требований, создание ER-диаграммы, нормализация.
- Шаг 4. Реализация: создание схемы в выбранной СУБД, загрузка данных, создание индексов и ограничений, написание запросов.
- Шаг 5. Анализ и тестирование: проверка целостности, тестирование запросов, оценка производительности.
- Шаг 6. Оформление и сдача: компоновка материалов, проверка на плагиат, финальная выверка.
- Шаг 7. Самоконтроль: заполнение чек-листа.

5) План-график и распределение времени (пример на семестр)

- Ранние недели: вводные занятия по методике работы, обзор СУБД, установка окружения.
- Средние недели: выполнение первых теоретических и практических заданий; создание базовой схемы.
- Средние недели: лабораторные работы with реальными данными; написание наборов запросов.
- Середина семестра: мини-проект (полная реализация базы, тестирование, отчёт).
- Конец семестра: итоговый проект/обзор технологий, защита результатов.
- Примерные сроки:
- Недели 1–2: требования, основы проектирования БД, первые задания.
- Недели 3–6: моделирование, создание схем, простые SQL-запросы.
- Недели 7–9: более сложные запросы, индексация, транзакции, хран. процедуры.
- Недели 10–12: мини-проект или комплексный проект, финальные отчёты, защита.

6) Оценивание и критерии контроля Общая шкала 0–100 баллов. Весовые категории можно адаптировать под учебное заведение.

- План и организация работы (10–15 баллов): наличие плана, этапы, сроки.
- Моделирование и нормализация (20–25 баллов): качество ER-модели, нормальные формы, объяснение выбора.
- Реализация базы (25–30 баллов): корректность DDL/DML, целостность данных, ограничения, индексы.
- Запросы и функциональность (20–25 баллов): полнота и оптимизация SQL-запросов, примеры отчётов.
- Оформление и стиль (10–15 баллов): единый стиль, полнота документации, отсутствие плагиата.
- Источники и оригинальность (5–10 баллов): корректная работа с источниками, оригинальные выводы.
- Приложения и коды (5–10 баллов): читаемость кода, комментарии, тестовые примеры.
- Рекомендация: приложите таблицу-рубрику с уровнями (отлично, хорошо, удовлетворительно, неудовлетворительно).

7) Ресурсы, инструменты и требования к технике

- СУБД: PostgreSQL, MySQL/MariaDB, SQLite, SQL Server, Oracle (выбор зависит от купца).
- Инструменты разработки: pgAdmin, DBeaver, MySQL Workbench, SQL Server Management Studio; IDE для скриптов.
- Язык запросов: SQL (ANSI SQL с особенностями выбранной СУБД).
- Наборы данных: Northwind, Sakila, AdventureWorks, открытые датасеты.
- Среда разработки: локальная машина или виртуальная машина/облачная лаборатория.
- Контроль версий: Git.
- Безопасность: обучение основам защиты данных, предотвращения SQL-инъекций, управление доступом.

8) Шаблоны документов (структура)

- Задание на самостоятельную работу по БД:
- Название работы
- Цели и задачи
- Методика выполнения
- Этапы выполнения (пошагово)
- Результаты
- Выводы
- Приложения (DDL/DML скрипты, схемы, данные)
- Источники

- Отчет по лабораторной работе:
- Титульный лист
- Цель и задачи
- Оборудование и ПО
- Ход выполнения и результаты
- Анализ результатов
- Выводы
- Приложения (коды, скрипты, таблицы)
- Список источников
- Мини-проект:
- Название проекта
- Цели
- ERD и описание схемы
- Реализация: DDL/DML, индексы, триггеры/процедуры
- Примеры запросов и отчётов
- Выводы
- Приложения и источники

Практическое занятие №1: «Создание базовых таблиц» в MS Access.

1) Цели и требования

- Освоить интерфейс MS Access: создание базы данных, проектирование таблиц в Design View, настройка полей и параметров.
- Научиться задавать первичные ключи и типы данных, выбирать подходящие данные для полей.
- Освоить создание отношений между таблицами и включение целостности ссылок (Referential Integrity).
- Получить навык заполнения тестовыми данными и проверить корректность соединений через простой SQL-запрос.
- Подготовить краткий отчёт: структура БД, описание полей, скриншоты и примеры запросов.

2) Необходимое ПО и окружение

- MS Access (часть пакета Microsoft 365/Office).
- Пример базы данных: можно начать с пустой базы и добавить таблицы вручную.
- Резервное копирование: желательно сделать копию файла базы данных перед началом работы.

3) Рекомендованная предметная область и структура БД (пример) Для наглядности можно использовать упрощённую БД библиотеки:

- Таблица Authors (авторы)
- AuthorID (Autonumber, первичный ключ)
- FirstName (Short Text, до 50)
- LastName (Short Text, до 50)
- Country (Short Text, до 50)
- BirthDate (Date/Time)
- Таблица Books (книги)
- BookID (Autonumber, первичный ключ)
- Title (Short Text, до 255)
- AuthorID (Long Integer) — внешний ключ на Authors.AuthorID
- Publisher (Short Text, до 100)
- PublishYear (Number, Integer)
- Таблица Members (пользователи/читатели)
- MemberID (Autonumber, первичный ключ)
- FirstName (Short Text, до 50)
- LastName (Short Text, до 50)
- Email (Short Text, до 100) — можно задать уникальный индекс
- Phone (Short Text, до 20)
- MemberSince (Date/Time)
- Таблица Loans (заимствования)
- LoanID (Autonumber, первичный ключ)
- BookID (Long Integer) — внешний ключ на Books.BookID
- MemberID (Long Integer) — внешний ключ на Members.MemberID
- LoanDate (Date/Time)
- ReturnDate (Date/Time)
- Status (Short Text, до 20) — напр., "In progress", "Returned"

4) Этапы занятия (пошагово)

- Шаг 0. Подготовка

- Создайте новую пустую базу данных: File > New > Blank Database; задайте имя (например, LibraryDB_Assignment.accdb) и выберите расположение.
- Шаг 1. Создание таблиц в Design View
- Authors
 - AuthorID: Data Type = AutoNumber, выберите Primary Key
 - FirstName: Short Text (50)
 - LastName: Short Text (50)
 - Country: Short Text (50)
 - BirthDate: Date/Time
- Books
 - BookID: AutoNumber, Primary Key
 - Title: Short Text (255)
 - AuthorID: Number/Long Integer (потом сделаем связь)
 - Publisher: Short Text (100)
 - PublishYear: Number (Integer)
- Members
 - MemberID: AutoNumber, Primary Key
 - FirstName: Short Text (50)
 - LastName: Short Text (50)
 - Email: Short Text (100)
 - Phone: Short Text (20)
 - MemberSince: Date/Time
- Loans
 - LoanID: AutoNumber, Primary Key
 - BookID: Number/Long Integer
 - MemberID: Number/Long Integer
 - LoanDate: Date/Time
 - ReturnDate: Date/Time
 - Status: Short Text (20)
- Шаг 2. Установка связей (Relationships)
- Откройте Relationships (Database Tools > Relationships).
- Добавьте все таблицы.
- Соедините AuthorID в Books с Authors.AuthorID. Установите Referential Integrity (Целостность ссылок). Включите каскадные обновления и каскадное удаление, если это требуется по условиям задачи.
- Соедините BookID в Loans с Books.BookID; соедините Members.MemberID с Loans.MemberID. Включите Referential Integrity.
- Шаг 3. Установка дополнительных ограничений
- В Members можно сделать уникальный индекс на Email (через Design View > внизу Indexes > создать новый индекс по полю Email с Unique = Yes).
- По желанию можно пометить некоторые поля как Required (например, Title, AuthorID в Books; Email и MemberID в Loans).
- Шаг 4. Ввод тестовых данных
- Введите несколько записей в Authors, затем в Books (указывайте корректный AuthorID), в Members и в Loans.
- Пример данных можно взять в качестве образца (сделайте небольшую порцию, чтобы проверить работу связей).
- Шаг 5. Проверка через простой запрос
- Создайте простой запрос (Query) на соединение:
 - Список книг с именем автора: `SELECT Books.Title, Authors.FirstName, Authors.LastName FROM Books INNER JOIN Authors ON Books.AuthorID = Authors.AuthorID;`
- Запустите запрос и убедитесь, что данные связаны корректно.
- Шаг 6. Сохранение и документация
- Сохраните базу данных.

- Сделайте скриншоты структуры таблиц (вид Design View) и Relationships, а также результатов запроса.
- Подготовьте короткий отчёт: какие поля созданы, какие связи заданы, какие тестовые данные введены.

5) Рекомендации по работе и типичные сложности

- Типы данных: используйте AutoNumber для ключей; используйте Long Integer (Number) для внешних ключей; строки — Short Text; даты — Date/Time.
- Совместимость ключей: внешние ключи должны состоять из того же типа, что и соответствующий первичный ключ.
- Целостность ссылок: всегда включайте Referential Integrity в Relationships; включение каскадных обновлений/удалений упрощает управление данными на практике.
- Уникальность и валидация: уникальный индекс на Email в Members помогает предотвратить дубли.
- Производительность: пока база небольшая, индексы на часто используемых полях (например, Email, Title) не обязательны, но полезны для практики.
- Безопасность и резервное копирование: регулярно сохраняйте копии базы данных, особенно перед большими изменениями.

6) Рекомендованный шаблон отчета по заданию

- Название работы: Практическое занятие №1. Создание базовых таблиц в MS Access
- Цели и задачи
- Описание среды и окружения
- Описание структуры БД (список таблиц и их полей)
- Описание ключевых ограничений и связей
- Ход выполнения (кратко по шагам)
- Результаты: скриншоты Design View и Relationships, примеры результатов запроса
- Выводы
- Приложения: скрипты или дампы данных (при желании)
- Источники

7) Пример минимального задания/книги для встраивания в курс

- Теоретическая часть: объяснить назначение первичных ключей и связей «один к многим» в контексте простых таблиц.
- Практическая часть: реализовать вышеописанный набор таблиц в MS Access, заполнить данными и выполнить один тестовый запрос.
- Дополнительно (для продвинутой части): добавить простую форму ввода данных для одной из таблиц или создать простой отчёт по таблице Books и Authors.

Практическое занятие №2 «Создание межтабличных связей»

1) Цели и требования

- Научиться строить межтабличные связи в MS Access через Relationships (Целостность ссылок).
- Освоить настройку отношений: один-ко-многим, многие-ко-многим через мостовую таблицу, а также включение каскадных обновлений/удалений при необходимости.
- Убедиться в корректности связи типов данных внешних ключей и первичных ключей родительских таблиц.
- Проверить работу связей через тестовые данные и простые запросы.

- Подготовить отчёт с иллюстрациями отношений и примерами запросов.

2) Необходимое ПО и окружение

- MS Access (часть Microsoft 365/Office).
- База данных из Практического занятия №1 либо новая база с базовыми таблицами Authors, Books, Members, Loans. Для второго занятия удобно добавить мостовую таблицу.
- Резервное копирование: рекомендуется перед началом сделать копию файла БД.

3) Предметная область и структура БД (пример) Продолжаем использовать библиотечную тематику. В дополнение к существующим таблицам можно ввести мостовую таблицу для связи книг и авторов.

- Таблица Authors
- AuthorID (Autonumber, PK)
- FirstName (Short Text)
- LastName (Short Text)
- Таблица Books
- BookID (Autonumber, PK)
- Title (Short Text)
- Publisher (Short Text)
- PublishYear (Number)
- Таблица BookAuthors (мостовая таблица для связи книг и авторов)
- BookID (Long Integer, FK на Books.BookID)
- AuthorID (Long Integer, FK на Authors.AuthorID)
- Первичный ключ: составной (BookID, AuthorID)
- Таблица Members и Таблица Loans — можно использовать из предыдущего занятия

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу данных из Практического занятия №1 или создайте новую.
- Убедитесь, что у таблиц Authors и Books корректные первичные ключи и подходящие типы данных для внешних ключей (BookAuthors.BookID и BookAuthors.AuthorID должны быть Long Integer).

Шаг 1. Добавление мостовой таблицы BookAuthors

- Перейдите в режим Design View и добавьте новую таблицу BookAuthors.
- Поля:
- BookID: Number (Long Integer)
- AuthorID: Number (Long Integer)
- Установите Composite Primary Key: выделите оба поля и нажмите Primary Key (поле будут помечены как PK вместе).
- Сохраните таблицу.

Шаг 2. Установка связей в Relationships

- Откройте Relationships (Database Tools > Relationships).
- Добавьте таблицы: Authors, Books, BookAuthors (и при желании Members, Loans).
- Создайте следующие связи:
- Authors.AuthorID 1—∞ BookAuthors.AuthorID
 - Включите Referential Integrity.

- При необходимости: Cascade Update Related Fields, Cascade Delete Related Records (осторожно: удаление может повлечь удаление связанных записей; используйте по необходимости).
- Books.BookID 1–∞ BookAuthors.BookID
 - Включите Referential Integrity.
 - При необходимости: Cascade Update Related Fields, Cascade Delete Related Records.
- При необходимости добавьте дополнительные связи между Books и Loans, Authors и Books (через BookAuthors вы получите многие–ко–многим сценарий).

Шаг 3. Проверка целостности и поведения связей

- Вставьте тестовые данные:
- Добавьте несколько Authors (например, 2–3 записи).
- Добавьте несколько Books (1–2 записи).
- Внесите пары в BookAuthors, связывая книги с авторами (одна книга может иметь несколько авторов, и один автор может писать несколько книг).
- Попробуйте:
- Ввести BookAuthors запись с несуществующим BookID — должна вылезти ошибка целостности.
- Удалить автора или книгу, имеющую связи в BookAuthors — если включены каскадные удаления, связанные записи в BookAuthors удалятся автоматически.
- Создайте простой запрос-JOIN, демонстрирующий работу связей (ниже в разделе “Запросы”).

Шаг 4. Запросы, демонстрирующие связи

- Запрос 1 (книги и их авторы): `SELECT Books.Title, Authors.FirstName, Authors.LastName FROM (Books INNER JOIN BookAuthors ON Books.BookID = BookAuthors.BookID) INNER JOIN Authors ON BookAuthors.AuthorID = Authors.AuthorID;`
- Запрос 2 (авторы и их книги): `SELECT Authors.FirstName, Authors.LastName, Books.Title FROM (Authors INNER JOIN BookAuthors ON Authors.AuthorID = BookAuthors.AuthorID) INNER JOIN Books ON BookAuthors.BookID = Books.BookID;`
- Запрос 3 (для одной книги вывести всех авторов): `SELECT Books.Title, Authors.FirstName, Authors.LastName FROM (Books INNER JOIN BookAuthors ON Books.BookID = BookAuthors.BookID) INNER JOIN Authors ON BookAuthors.AuthorID = Authors.AuthorID WHERE Books.Title = 'Название книги';`

Шаг 5. Расширения и полезности

- В Bridge (BookAuthors) можно использовать больше функций: добавить уникальный индекс на пару полей, если нужно предотвратить дубликаты пар.
- Можно добавить формы/отчёты, которые показывают состав авторов по книгам, используя связанный набор данных.

5) Советы и возможные сложности

- Типы данных: внешний ключ в мостовой таблице должен совпадать по типу с PK родительских таблиц (BookID и AuthorID — Long Integer).
- Composite PK: в Access можно задать составной первичный ключ на BookAuthors (выделить оба поля в Design View и нажать Primary Key).
- Целостность ссылок: обязательно включайте Referential Integrity; будьте осторожны с Cascade Delete — он может удалить множество записей в зависимых таблицах.
- Производительность: для небольших наборов данных всё работает быстро; при росте объёмов можно рассмотреть индексы на полях BookID и AuthorID в мостовой таблице.
- Безопасность и резервное копирование: регулярно сохраняйте резервные копии БД, особенно перед крупными изменениями.

6) Рекомендованный формат отчета по занятию

- Название занятия: Практическое занятие №2. Создание межтабличных связей в MS Access
- Цели и задачи
- Среда и окружение
- Описание структуры БД (таблицы Authors, Books, BookAuthors и т.д.)
- Описание связей и правил целостности
- Ход выполнения (пошагово)
- Результаты: скриншоты Relationships, примеры запросов и их результаты
- Выводы
- Приложения: дампы данных, SQL-скрипты (если применимо)

Практическое занятие №3 «Создание запроса на выборку»

1) Цели и требования

- Освоить создание выборочных запросов в MS Access: простые и многотабличные запросы с различными условиями.
- Научиться использовать соединения (INNER JOIN, LEFT JOIN) для получения данных из нескольких таблиц.
- Освоить фильтрацию данных (WHERE), сортировку (ORDER BY), вычисляемые поля и функции.
- Овладеть агрегатами (GROUP BY, COUNT, SUM и т. д.) и простыми подзапросами/параметрическими запросами.
- Подготовить короткий отчёт с примерами запросов и результатами их выполнения.

2) Необходимое ПО и окружение

- MS Access (часть Microsoft 365/Office).
- База данных из Практического занятия №1 (или новая база с таблицами Authors, Books, BookAuthors, Members, Loans).
- Резервное копирование: желательно сделать копию базы перед началом работы.

3) Предметная область и структура БД (пример) Используем ту же библиотечную тематику:

- Таблицы: Authors, Books, BookAuthors (мостовая для связи многие–ко–многим), Members, Loans.
- Пример ключевых полей:
- Authors: AuthorID (PK), FirstName, LastName
- Books: BookID (PK), Title, Publisher, PublishYear
- BookAuthors: BookID, AuthorID (плотный составной PK: BookID + AuthorID)
- Members: MemberID (PK), FirstName, LastName, Email, etc.
- Loans: LoanID (PK), BookID, MemberID, LoanDate, ReturnDate, Status

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу данных из Практического занятия №1 или создайте новую.
- Убедитесь, что ключи и типы данных соответствуют внешним ключам (BookAuthors.BookID и BookAuthors.AuthorID должны быть Long Integer).

Шаг 1. Планирование и работа с запросами в Design View

- Откройте Create > Query Design.
- Добавьте нужные таблицы (Books, Authors, BookAuthors, Members, Loans).
- Разместите поля, которые будете использовать в запросе.
- При необходимости переключитесь в SQL View для редактирования SQL напрямую.

Шаг 2. Примеры запросов на выборку

Пример 1. Список книг с годом публикации и издателем

- Цель: получить базовую информацию о книгах.
- SQL: `SELECT Books.Title, Books.PublishYear, Books.Publisher FROM Books ORDER BY Books.PublishYear DESC, Books.Title;`

Пример 2. Список книг и их авторов (многотабличное соединение)

- Цель: показать книги вместе с именами авторов через мостовую таблицу.
- SQL: `SELECT Books.Title, Authors.FirstName, Authors.LastName FROM (Books INNER JOIN BookAuthors ON Books.BookID = BookAuthors.BookID) INNER JOIN Authors ON BookAuthors.AuthorID = Authors.AuthorID ORDER BY Books.Title, Authors.LastName, Authors.FirstName;`

Пример 3. Список читателей и их займов (состояние и даты)

- Цель: увидеть, какие книги взяты читателями и даты займов.
- SQL: `SELECT Members.FirstName, Members.LastName, Books.Title AS BookTitle, Loans.LoanDate, Loans.ReturnDate, Loans.Status FROM (Members INNER JOIN Loans ON Members.MemberID = Loans.MemberID) INNER JOIN Books ON Loans.BookID = Books.BookID ORDER BY Members.LastName, Members.FirstName, Loans.LoanDate;`

Пример 4. Поиск займов в заданный диапазон дат (параметрический запрос)

- Цель: выбирать займы, попавшие в указанный период.
- SQL: `SELECT Members.FirstName, Members.LastName, Books.Title, Loans.LoanDate, Loans.ReturnDate FROM (Members INNER JOIN Loans ON Members.MemberID = Loans.MemberID) INNER JOIN Books ON Loans.BookID = Books.BookID WHERE Loans.LoanDate BETWEEN [StartDate] AND [EndDate] ORDER BY Loans.LoanDate;`

Пример 5. Сводная информация: число займов по каждому читателю

- Цель: агрегировать данные по читателям.
- SQL: `SELECT Members.MemberID, Members.FirstName, Members.LastName, COUNT(Loans.LoanID) AS LoansCount FROM Members LEFT JOIN Loans ON Members.MemberID = Loans.MemberID GROUP BY Members.MemberID, Members.FirstName, Members.LastName ORDER BY LoansCount DESC;`

Пример 6. Поиск книг по названию с использованием подстановки

- Цель: демонстрация фильтрации текста.
- В Access используется символ подстановки * (не %).
- SQL: `SELECT Title, Publisher, PublishYear FROM Books WHERE Title LIKE "Data" ORDER BY Title;`

Пример 7. Простейшее объединение авторов и их книг (много-ко-многим)

- Цель: показать пару таблиц через мостовую.

- SQL: SELECT Authors.FirstName, Authors.LastName, Books.Title FROM (Authors INNER JOIN BookAuthors ON Authors.AuthorID = BookAuthors.AuthorID) INNER JOIN Books ON BookAuthors.BookID = Books.BookID ORDER BY Authors.LastName, Authors.FirstName, Books.Title;

Шаг 3. Тестирование и проверка результатов

- Выполните каждый запрос и проверьте соответствие ожиданиям.
- Попробуйте изменить критерии отбора, порядок сортировки и набор полей.
- Убедитесь, что запросы корректно работают со случаем отсутствия связей (LEFT JOIN). Например, читатель без займов должен все равно появляться в запросе, если используете LEFT JOIN.

Шаг 4. Сохранение, документирование и повторяемость

- Сохраните каждый запрос под понятным именем (Query1_Books, Query2_BooksAuthors, etc.).
- Сделайте скриншоты полученных результатов и структуры запроса (в Design View/SQL View).
- Подготовьте небольшой отчет с описанием цели каждого запроса и полученными данными.

5) Советы и потенциальные сложности

- Access с Wildcards: помните, что в MS Access для LIKE используются *. Для одиночных символов — ?.
- Типы данных: внешние ключи должны совпадать по типу с первичными ключами родительских таблиц.
- Эффективность: если база большая, разумно добавить индексы на часто фильтруемые поля (Title, PublishYear, LastName и т. д.).
- Права и целостность: при использовании JOIN-условий следите за корректной логикой соединений, особенно при LEFT JOIN.
- Переход между Design View и SQL View поможет ученикам увидеть связь между визуальными действиями и реальным SQL.

6) Рекомендованный формат отчета по занятию

- Название занятия: Практическое занятие №3. Создание запроса на выборку в MS Access
- Цели и задачи
- Окружение и база данных
- Описание рассмотренных запросов (с указанием целей и ключевых SQL-выражений)
- Результаты: скриншоты результатов выполнения запросов
- Выводы и замечания
- Приложения: копии SQL-кода, дампы данных, примеры ответов

Практическое занятие №4 «Создание запроса «с параметром»

1) Цели и требования

- Освоить создание выборочных запросов с параметрами (пользовательский ввод во время выполнения).
- Научиться применять простые и многотабличные параметры, диапазоны дат, текстовые и числовые параметры.

- Ознакомиться с двумя способами задания параметров: через явное выражение PARAMETERS в SQL и через визуальные промпты в Design View.
- Научиться связывать параметры с формами (Forms) для интерактивного ввода.
- Подготовить небольшой отчёт с примерами запросов и результатами.

2) Необходимое ПО и окружение

- MS Access (часть Microsoft 365/Office).
- База данных из Практического занятия №1 (или актуальная база с Authors, Books, BookAuthors, Members, Loans).
- Резервное копирование базы перед занятиями.

3) Структура БД (пример) Используем тематику библиотеки:

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK состоит из BookID+AuthorID
- Members(MemberID PK, FirstName, LastName, Email, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или создайте новую.
- Убедитесь, что структура таблиц и типы данных корректны для параметрических запросов.

Шаг 1. Создание параметрического запроса через Design View

- Откройте Create > Query Design.
- Добавьте нужные таблицы (например, Books, Authors, BookAuthors).
- Переключитесь в SQL View для редактирования SQL напрямую, либо используйте поле Criteria в Design View, чтобы Access автоматически добавил параметр в WHERE.
- В любом случае цель — чтобы запрос запрашивал значение у пользователя во время выполнения.

Шаг 2. Примеры запросов с параметрами (SQL)

Пример 1. Поиск книг по названию с параметром

- Цель: найти книги по частичному названию.
- SQL (явное объявление параметра): `PARAMETERS [pTitle] TEXT; SELECT Books.Title, Books.Publisher, Books.PublishYear FROM Books WHERE Books.Title LIKE [pTitle] & "*" ORDER BY Books.Title;`

Пример 2. Диапазон лет издания (параметрический диапазон)

- Цель: выбрать книги в заданном диапазоне лет.
- SQL: `PARAMETERS [pYearMin] SHORT, [pYearMax] SHORT; SELECT Title, PublishYear, Publisher FROM Books WHERE PublishYear BETWEEN [pYearMin] AND [pYearMax] ORDER BY PublishYear;`

Пример 3. Поиск займов по диапазону дат (параметры даты)

- Цель: узнать займы в заданном периоде.

- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Members.FirstName, Members.LastName, Books.Title, Loans.LoanDate, Loans.ReturnDate FROM (Members INNER JOIN Loans ON Members.MemberID = Loans.MemberID) INNER JOIN Books ON Loans.BookID = Books.BookID WHERE Loans.LoanDate BETWEEN [pStartDate] AND [pEndDate] ORDER BY Loans.LoanDate;

Пример 4. Поиск по статусу займа и фамилии клиента (несколько параметров)

- Цель: фильтр по статусу займа и частичной фамилии клиента.
- SQL: PARAMETERS [pStatus] TEXT, [pLastName] TEXT; SELECT Members.FirstName, Members.LastName, Loans.Status, Books.Title FROM (Members INNER JOIN Loans ON Members.MemberID = Loans.MemberID) INNER JOIN Books ON Loans.BookID = Books.BookID WHERE Loans.Status = [pStatus] AND Members.LastName LIKE [pLastName] & "*" ORDER BY Members.LastName, Members.FirstName;

Пример 5. Комбинация по авторам и годам выпуска (мостовая таблица)

- Цель: книги какого-либо автора, выпущенные в диапазон лет.
- SQL: PARAMETERS [pAuthorLast] TEXT, [pYearMin] SHORT, [pYearMax] SHORT; SELECT Books.Title, Authors.LastName, Books.PublishYear FROM (Books INNER JOIN BookAuthors ON Books.BookID = BookAuthors.BookID) INNER JOIN Authors ON BookAuthors.AuthorID = Authors.AuthorID WHERE Authors.LastName LIKE [pAuthorLast] & "*" AND Books.PublishYear BETWEEN [pYearMin] AND [pYearMax] ORDER BY Authors.LastName, Books.Title;

Пример 6. Простой параметр через форму (использование контролов формы)

- Цель: ввод параметров через форму, затем выполнение запроса.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Members.FirstName, Members.LastName, Loans.LoanDate FROM Loans INNER JOIN Members ON Loans.MemberID = Members.MemberID WHERE Loans.LoanDate BETWEEN [Forms]![frmLoansSearch]![dtpStart] AND [Forms]![frmLoansSearch]![dtpEnd] ORDER BY Loans.LoanDate;

Примечания по работе с LIKE в Access:

- В Access подстановочный знак — *. Один символ — ?.
- В примерах выше для частичных совпадений применяется LIKE [параметр] & "*".

Шаг 3. Запуск и ввод параметров

- Выполните каждый запрос. Access покажет диалог prompt для параметра.
- Введите требуемые значения или подключите параметры к полям формы (для примера 6).

Шаг 4. Расширения и практические задания

- Создать 2–3 параметрических запроса на разные темы (книги по диапазону лет, займы по дате, поиск авторов по фамилии и т. д.).
- Реализовать форму поиска займов (frmLoansSearch) с полями dtpStart, dtpEnd, cmbStatus и привязкой к соответствующим параметрическим запросам.
- Применить параметрические запросы в отчетах (например, вывод по займам за выбранный период).

Шаг 5. Тестирование и документация

- Протестируйте параметры с разными значениями, проверьте корректность working with NULLs (где BY DEFAULT параметры требуют значение; можно использовать альтернативные критерии или обработку NULL).
- Сохраните каждый запрос под понятным именем.
- Сделайте скриншоты панели Design View и результатов выполнения запросов.
- Подготовьте короткий отчёт: цель запроса, параметры, примеры результатов.

5) Советы и потенциальные сложности

- Явно объявляйте PARAMETERS, чтобы избежать автоматических промптов и обеспечить ясность.
- Параметры в Access чувствительны к типам данных: TEXT, DateTime, Short (число). Убедитесь в корректности типов.
- При использовании дат учитывайте локализацию форматов (скобки DateTime).
- Если используете параметр через форму, охраняйте ссылки на форму/контролы корректными именами.
- При необходимости используйте HAS HAVING для агрегатов в параметрических условиях (примерно как в предыдущем занятии).

6) Формат отчета по занятию

- Название занятия: Практическое занятие №4. Создание запроса «с параметром» в MS Access
- Цели и задачи
- Окружение и база данных
- Описание рассмотренных примеров параметрических запросов (с указанием SQL)
- Результаты: скриншоты выполнения запросов
- Выводы и замечания
- Приложения: копии SQL-кода, дампы данных, примеры форм (frmLoansSearch) и параметрических сценариев

7) Примеры заданий для студентов

- Реализовать 2–3 параметрических запроса по теме библиотеки (книги по названию, займы по диапазону дат, авторы по фамилии).
- Сконструировать простую форму поиска займов и связать её с параметрическими запросами.
- Подготовить небольшой отчёт с результатами тестирования.

Практическое занятие №5 Создание итогового запроса

1) Цели и требования

- Сформировать один или несколько итоговых запросов, демонстрирующих владение приемами конструирования сложных выборок.
- Использовать соединения (INNER JOIN, LEFT JOIN), агрегатные функции (COUNT, SUM, AVG, MAX, MIN), группировку (GROUP BY) и условия HAVING.
- Применять подзапросы, подстановочные параметры и, при желании, Crosstab-запросы.
- Связать запрос с формами/отчётами для интерактивного ввода и представления результатов.
- Подготовить краткий отчёт с описанием цели итогового запроса и результатов выполнения.

2) Необходимое ПО и окружение

- MS Access (часть Microsoft 365/Office).

- База данных из Практического занятия №1 (Authors, Books, BookAuthors, Members, Loans) или эквивалентная с теми же таблицами.
- Резервная копия базы перед занятиями.

3) Структура БД (для напоминания)

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из Практического занятия №1 или создайте новую/переименуйте для итогового занятия.
- Проверьте корректность связей и типов данных, подготовьте даты и статусы для тестирования.

Шаг 1. Проектирование итогового запроса

- Откройте Create > Query Design.
- Добавьте нужные таблицы (например, Books, Authors, BookAuthors, Loans, Members).
- Определите набор полей и критериев, которые помогут построить «итог» (агрегаты, группы, фильтры).

Шаг 2. Примеры итоговых запросов (SQL) Пример 1. Итоговые займы по читателям за заданный период

- Цель: получить список читателей с количеством займов и средним временем владения книгами за период.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Members.MemberID, Members.FirstName, Members.LastName, COUNT(Loans.LoanID) AS LoansCount, AVG(DateDiff("d", Loans.LoanDate, Nz(Loans.ReturnDate, Date()))) AS AvgLoanDays FROM Members LEFT JOIN Loans ON Members.MemberID = Loans.MemberID WHERE Loans.LoanDate BETWEEN [pStartDate] AND [pEndDate] GROUP BY Members.MemberID, Members.FirstName, Members.LastName ORDER BY LoansCount DESC;

Пример 2. Топ-5 книг по спросу за период (книги с наибольшим числом займов)

- Цель: определить книги с наибольшим спросом.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Books.Title, Authors.LastName, Authors.FirstName, COUNT(Loans.LoanID) AS LoansCount FROM (((Books INNER JOIN BookAuthors ON Books.BookID = BookAuthors.BookID) INNER JOIN Authors ON BookAuthors.AuthorID = Authors.AuthorID) INNER JOIN Loans ON Books.BookID = Loans.BookID) WHERE Loans.LoanDate BETWEEN [pStartDate] AND [pEndDate] GROUP BY Books.Title, Authors.LastName, Authors.FirstName ORDER BY LoansCount DESC, Books.Title;

Пример 3. Сводная таблица займов по статусу и месяцу (Crosstab)

- Цель: сводка займов по статусу по каждому месяцу.
- Примечание: Crosstab может потребовать подготовки выражений для периода.
- SQL (примерный шаблон, адаптируйте под формат даты): TRANSFORM Count(Loans.LoanID) AS CountOfLoans SELECT DatePart("m", Loans.LoanDate) AS MonthOfLoan FROM Loans

GROUP BY DatePart("m", Loans.LoanDate) ORDER BY DatePart("m", Loans.LoanDate) PIVOT Loans.Status;

Пример 4. Итог по авторам: количество книг и суммарное число займов за период

- Цель: увидеть активность авторов за период.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Authors.AuthorID, Authors.FirstName, Authors.LastName, COUNT(DISTINCT Books.BookID) AS BooksCount, COUNT(Loans.LoanID) AS LoansCount FROM ((Authors LEFT JOIN BookAuthors ON Authors.AuthorID = BookAuthors.AuthorID) LEFT JOIN Books ON BookAuthors.BookID = Books.BookID) LEFT JOIN Loans ON Books.BookID = Loans.BookID WHERE Loans.LoanDate BETWEEN [pStartDate] AND [pEndDate] GROUP BY Authors.AuthorID, Authors.FirstName, Authors.LastName ORDER BY LoansCount DESC;

Пример 5. Итоговый запрос с подзапросом: книги, которых не хватало в период

- Цель: найти книги, для которых за период не было займов.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Books.Title FROM Books WHERE Books.BookID NOT IN (SELECT Loans.BookID FROM Loans WHERE Loans.LoanDate BETWEEN [pStartDate] AND [pEndDate]) ORDER BY Books.Title;

Пример 6. Простой итоговый запрос через форму (показ результатов на форме)

- Цель: вывод итогов по параметрам, введенным через форму.
- SQL: PARAMETERS [pStartDate] DateTime, [pEndDate] DateTime; SELECT Members.FirstName, Members.LastName, COUNT(Loans.LoanID) AS LoansCount FROM Members LEFT JOIN Loans ON Members.MemberID = Loans.MemberID WHERE Loans.LoanDate BETWEEN [Forms]![frmLoansSummary]![dtpStart] AND [Forms]![frmLoansSummary]![dtpEnd] GROUP BY Members.FirstName, Members.LastName ORDER BY LoansCount DESC;

Замечания по примерам:

- Для LIKE-подстановок можно использовать wildcard в Access: * вместо % и ? для одного символа.
- Для корректной работы с NULL используйте Nz(...) при расчётах дат и значений.
- При использовании Crosstab нужно внимательно подбирать поля в ROW, COLUMN и VALUE.

Шаг 3. Реализация и тестирование

- Создайте 2–4 итоговых запроса на разные темы (например, займы по периоду, активность читателей, популярные книги).
- При желании подключите параметры к формам для интерактивности (Form frmLoansSummary, dtpStart, dtpEnd и т.д.).
- Протестируйте запросы на разных диапазонах дат и статусах.

Шаг 4. Документация и представление результатов

- Сохраните каждый запрос под понятным именем (например, qryLoansByPeriod, qryTopBooksByLoans, qryAuthorsActivity, и т.д.).
- Сделайте скриншоты SQL View и результата выполнения.
- Подготовьте компактный отчёт: цель итога, структура запроса, параметры, примеры результатов.

5) Советы и возможные сложности

- В итоговых запросах разумно сочетать агрегаты, группировку и HAVING (например, HAVING COUNT(Loans.LoanID) > 2).
- Для дат используйте параметры типа DateTime и корректно задавайте диапазоны.
- Если хотите, чтобы итоговый запрос был максимально наглядным, попробуйте объединить данные используя несколько уровней соединений (LEFT JOIN для сохранения «пустых» записей).
- Crosstab-запросы хороши для сводок, но требуют аккуратной настройки.

6) Формат отчета по занятию

- Название занятия: Практическое занятие №5. Создание итогового запроса в MS Access
- Цели и задачи
- Окружение и база данных
- Описание рассмотренных итоговых запросов (со ссылками на SQL)
- Результаты: скриншоты выполнения запросов и схемы соединений
- Выводы и замечания
- Приложения: копии SQL-кода, дампы данных, примеры форм/отчетов

Практическое занятие №6 Создание вычисляемого запроса

1) Цели и требования

- Освоить создание запросов, в которых часть полей вычисляется на лету (вычисляемые поля).
- Изучить простые и сложные выражения: конкатенацию строк, арифметические расчёты, работу с датами, условия (If), обработку NULL (Nz).
- Научиться добавлять вычисляемые поля как в SQL, так и через Design View (поле Calculated).
- Продемонстрировать применение вычисляемых полей в отчетах и формах.
- Подготовить краткий отчёт с примерами вычисляемых запросов и результатов.

2) Необходимое ПО и окружение

- MS Access (Office 365/Microsoft 365 или локальная версия).
- База данных из Практического занятия №1 (Authors, Books, BookAuthors, Members, Loans) или эквивалентная с теми же таблицами.
- Резервная копия базы перед занятиями.

3) Структура БД (напоминание)

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, BirthDate, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или создайте копию для вычисляемого занятия.
- Убедитесь, что есть поля, подходящие для вычислений (BirthDate в Members, PublishYear в Books, LoanDate/ReturnDate в Loans).

Шаг 1. Создание вычисляемого поля в Design View

- Откройте Create > Query Design.
- Добавьте нужные таблицы (например, Members, Loans, Books, Authors).
- В пустом поле SELECT введите выражение и имя поля: например FullName: [FirstName] & " " & [LastName]
- Переключитесь в SQL View, чтобы увидеть итоговый SQL, или сохраняйте как расчётное поле в Design View.
- Выражения можно строить через Expression Builder (Правая кнопка мыши в поле Calculated или кнопка Build).

Шаг 2. Примеры вычисляемых полей и соответствующий SQL

Пример 1. Конкатенация имени и фамилии читателя

- Цель: вывести полное имя читателя.
- SQL: FullName: [FirstName] & " " & [LastName] (или в SQL-запросе: SELECT Members.FirstName & " " & Members.LastName AS FullName ...)

Пример 2. Возраст читателя по BirthDate

- Цель: вычислить возраст на текущий момент.
- SQL: Age: DateDiff("yyyy", [BirthDate], Date()) - IIf(DateSerial(Year(Date()), Month([BirthDate]), Day([BirthDate])) > Date(), 1, 0) или проще: Age: DateDiff("yyyy", [BirthDate], Date()) - IIf(Month([BirthDate]) > Month(Date()) Or (Month([BirthDate]) = Month(Date()) And Day([BirthDate]) > Day(Date())), 1, 0)
- Примечание: можно использовать более простой вариант с Nz и проверками на NULL.

Пример 3. Дни с момента займа (для займов, вернуть или еще не вернуть)

- Цель: определить продолжительность займа на дату возврата или текущую дату, если возврат не выполнен.
- SQL: DaysLoan: DateDiff("d", [LoanDate], Nz([ReturnDate], Date()))
- Это вычисляемое поле можно использовать в агрегациях или фильтрах.

Пример 4. Суммарный штраф за просрочку

- Цель: рассчитать штраф на основе количества дней просрочки (например, 0.5 у.е. за каждый просроченный день после 14 дней).
- SQL: Fine: IIf([Status]="On loan", IIf(DateDiff("d", [LoanDate], Nz([ReturnDate], Date())) > 14, (DateDiff("d", [LoanDate], Nz([ReturnDate], Date())) - 14) * 0.5, 0), 0)
- Примечание: можно адаптировать коэффициенты и порог просрочки под вашу политику.

Пример 5. Длина названия книги

- Цель: вычислить длину заголовка книги.
- SQL: TitleLength: Len([Title])

Пример 6. Агрегаты с вычисляемыми полями (например, средняя продолжительность займа по читателю)

- Цель: по каждому читателю посчитать среднюю длительность займов.
- SQL (похожий на обычную агрегацию, добавляя DaysLoan как вычисляемое поле): SELECT Members.MemberID, Members.FirstName, Members.LastName, AVG(DateDiff("d", Loans.LoanDate, Nz(Loans.ReturnDate, Date()))) AS AvgDaysLoan FROM Members LEFT JOIN

Loans ON Members.MemberID = Loans.MemberID GROUP BY Members.MemberID,
Members.FirstName, Members.LastName;

Пример 7. Вычисляемое поле с условием (IsOverdue)

- Цель: пометить просроченные займы.
- SQL: IsOverdue: If([Status]="On loan" And DateDiff("d", [LoanDate], Nz([ReturnDate], Date())) > 14, True, False)

Шаг 3. Реализация и тестирование

- Реализуйте 2–4 вычисляемых запроса на разные темы (читатели и их возраст, займы и их продолжительность, популярность книг по длине названия и пр.).
- Попробуйте как простые вычисления (конкатенации, Len, DateDiff), так и более сложные с Nz, If.
- Протестируйте на разных диапазонах дат и на пустых значениях (NULL).

Шаг 4. Связь с формами и отчетами

- Добавьте вычисляемые поля в форму/отчет, чтобы пользователи видели результат прямо в интерфейсе.
- Например, форму frmLoansSummary можно дополнить полем FullName и AvgDaysLoan для удобства просмотра.

Шаг 5. Документация и представление результатов

- Сохраните каждый вычисляемый запрос под понятным именем (qryReadersWithAges, qryLoansWithDurations, qryBookTitleLengths и т.д.).
- Сделайте скриншоты Design View и результатов выполнения.
- Подготовьте краткий отчет: цель вычисляемого запроса, используемые выражения, примеры результатов.

5) Советы и возможные сложности

- Используйте Nz() для обработки NULL в вычисляемых полях, чтобы запросы не падали.
- В выражениях аккуратно работайте с датами: Date(), DateDiff, DateSerial, и т. п.
- В Design View используйте Expression Builder, чтобы формулы не содержали синтаксических ошибок.
- При сложных вычислениях разбивайте логику на несколько вычисляемых полей, чтобы облегчить отладку.
- Вычисляемые поля могут использоваться в отчетах и формах так же, как и обычные поля.

6) Формат отчета по занятию

- Название занятия: Практическое занятие №6. Создание вычисляемого запроса в MS Access
- Цели и задачи
- Окружение и база данных
- Описание рассмотренных примеров вычисляемых полей (со ссылками на SQL)
- Результаты: скриншоты Design View и запросов
- Выводы и замечания
- Приложения: копии SQL-кода, примеры форм/отчетов с вычисляемыми полями

Практическое занятие №7 Создание формы с помощью Мастера

1) Цели и требования

- Научиться быстро создать удобную форму с подформой (master-detail) с помощью Мастера форм.
- Освоить автоматическую настройку связи между формой и подформой (Link Master Field / Link Child Field).
- Ознакомиться с настройками формы: режим ввода данных, внешний вид, навигация, стиль.
- Сделать форму пригодной для ввода данных и просмотра связанных записей (например, займы конкретного читателя или книги).
- Подготовить небольшой отчёт о снятых результатах и сделанных настройках.

2) Необходимое ПО и окружение

- MS Access (Office 365/Microsoft 365 или локальная версия).
- База данных из Практического занятия №1 (Authors, Books, BookAuthors, Members, Loans) или эквивалентная с теми же таблицами.
- Резервная копия базы перед занятием.

3) Напоминание о структуре БД

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, BirthDate, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или создайте копию под вычисления/формы.
- Убедитесь, что задействованы подходящие поля для формы: для Members — MemberID, FirstName, LastName; для Loans — LoanID, MemberID, BookID, LoanDate, ReturnDate; для Books — BookID, Title и т.д.
- Проверьте наличие отношений между таблицами (Master-Child) для корректной работы подформ.

Шаг 1. Запуск Мастера форм

- Вкладка: Create (Создать) → Form Wizard (Мастер форм).
- В списке таблиц/запросов выберите главную таблицу для формы. Рекомендации по выбору:
- Простая форма с подформой: выбрать Members как главную таблицу, связанная подформа — Loans.
- Альтернатива: выбрать Books как главную таблицу, связанная подформа — BookAuthors (или Loans, если вы хотите видеть займы по книге).
- Выбор полей: возьмите основные поля для главной формы (например, для Members: MemberID, FirstName, LastName, Email). Затем выберите поля подформы (например, Loans: LoanID, LoanDate, ReturnDate, Status, BookID).

Шаг 2. Выбор подформы

- Мастер форм предложит создать подформу, если найдёте связанную таблицу. Выберите Loans как связанную таблицу.

- В следующем шаге мастер предложит способ отображения подформы (например, табличная или листовая). Выберите удобный вариант (Columnar/datasheet).
- Мастер автоматически предложит связи между полями: Link Master Field = MemberID (из главной формы) и Link Child Field = MemberID (из подформы Loans). Это обеспечит синхронизацию.

Шаг 3. Выбор макета и стиля

- Выберите желаемый макет формы (Columnar, Tabular, Datasheet) и стиль оформления.
- По завершении мастер создаст две связанные формы: главную и подформу (включая связь между ними).

Шаг 4. Завершение и базовая настройка

- Нажмите Finish. Откроется форма в режиме Design View или Form View в зависимости от версии.
- При необходимости перейдите в Form Design, чтобы выполнить небольшие настройки:
- Установить режим редактирования данных для главной формы (Data Entry): Да/Нет.
- Установить параметры подформы (например, Allow Additions, Scroll Bars, и т.д.).
- Убедитесь, что поля чтения и записи корректно работают, особенно ключевые поля (MemberID, LoanID).
- Включите удобные кнопки навигации и сохранения, при необходимости добавьте кнопки для добавления нового займа.

Шаг 5. Тестирование

- Переключитесь в Form View.
- Выберите существующего читателя и проверьте, отображаются его займы во подформе.
- Попробуйте добавить новый займ через подформу (создание новой записи в Loans с предустановленным MemberID).
- Попробуйте выбрать книгу в подформе (если подформа включает BookID) и заполнить даты займа.

Шаг 6. Расширения (по желанию)

- Добавьте дополнительные элементы управления: например, комбобоксы для выбора читателя и книги в главной форме или подформе.
- Добавьте в главную форму кнопку «Новый заем» для быстрого перехода к созданию записи в Loans.
- При желании создайте второй уровень подформы: например, в Loans есть детали книги — создать вторую вложенную подформу на BookDetails (если нужно).

Шаг 7. Документация и представление результатов

- Сохраните форму под понятным именем, например:
- frmMemberLoans (главная форма Members с подформой Loans)
- frmBooksAuthors (главная форма Books с подформой BookAuthors) — если используете мостовую таблицу.
- Сделайте скриншоты: Design View для формы, Form View с данными.
- Подготовьте краткий отчет: цель формы, структура связей, как настраивалась связь Master-Child, примеры сценариев использования.

5) Советы и возможные сложности

- Убедитесь, что между главной и подформой правильно настроены поля связи (Link Master Field / Link Child Field по соответствующим ключам, например MemberID или BookID).
- Для подформы в режиме редактирования включите Allow Additions, чтобы можно было добавлять новые записи напрямую во вложенной таблице.
- Если подформа не отображает данные, проверьте активность Relationship и корректность полей связи.
- В некоторых случаях целесообразно заменить простую подформу на подформу на запросе, чтобы показать нужные поля (например, отображать заголовок книги вместо BookID).

6) Формат отчёта по занятию

- Название занятия: Практическое занятие №7. Создание формы с помощью Мастера в MS Access
- Цели и задачи
- Окружение и база данных
- Этапы создания формы через Мастер форм (с пояснениями по каждому шагу)
- Результаты: сохранённые формы, скриншоты Form View и Design View
- Выводы и замечания
- Приложения: инструкции к выбору полей, ссылки на демонстрационные сценарии

Практическое занятие №8 Создание формы в режиме Конструктора

1) Цели и требования

- Освоить создание формы с нуля в Design View, управлять источником данных и расположением элементов.
- Научиться добавлять и связывать элементы управления вручную (полей, меток, кнопок, подформ).
- Научиться настраивать свойства формы (режим ввода, навигацию, стиль, формат).
- Разобрать сценарий мастер-деталь: форма главная с подформой, связь Master-Child через соответствующие поля.
- Подготовить краткий отчёт: структура формы, используемые свойства, примеры взаимодействий.

2) Необходимое ПО и окружение

- MS Access (Office 365/Microsoft 365 или локальная версия).
- База данных из Практического занятия №1 (Authors, Books, BookAuthors, Members, Loans) или эквивалентная с теми же полями.
- Резервная копия базы перед занятием.

3) Напоминание о структуре БД

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, BirthDate, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или сделайте копию.
- Убедитесь, что есть ключевые поля для формы: Members (MemberID, FirstName, LastName, Email), Loans (LoanID, MemberID, BookID, LoanDate, ReturnDate, Status), Books (BookID, Title).

Шаг 1. Создание формы в Конструкторе (Design View)

- Откройте Create > Design View (Конструктор).
- Выберите источник данных формы: кликните правой кнопкой на пустом поле и задайте Record Source (например, Members или Loans, если вы хотите форму главная).
- Установите размер и расположение рабочей области.

Шаг 2. Добавление полей вручную

- В панели Design, выберите Add Existing Fields (или используйте инструмент TextBox/Label из панели инструментов).
- Добавляйте нужные поля: например, для формы Members добавьте TextBox для MemberID, FirstName, LastName, Email.
- Подпишите метки (Label) и привяжите TextBox к полю через свойство Control Source.
- При необходимости добавьте вычисляемые поля (напр., FullName через выражение [FirstName] & " " & [LastName]) как Calculated Control (TextBox с Control Source = "FullName: ..." в поле).

Шаг 3. Настройка подформы (master-detail)

- В Design View добавьте подформу (Subform/Subreport) с помощью Subform/Subreport control.
- Выберите источник подформы (обычно Loans, если главная форма — Members).
- Установите Link Master Fields и Link Child Fields (например, MemberID).
- Разместите подформу так, чтобы она отображала связанные записи.

Шаг 4. Настройка свойств формы

- Откройте свойства формы (Property Sheet).
- Set Data tab:
- Record Source: укажите таблицу/запрос.
- Data Entry: Yes/No, если нужна только запись новых данных.
- Allow Edits, Allow Deletions, Allow Additions: по необходимости.
- Настройте оформление: Border Style, AutoFormat, Font, Scheme, Form Header/Footer.
- Включите навигацию (Record selectors, Navigation) по желанию.

Шаг 5. Добавление кнопок и базовая навигация

- Добавьте на форму кнопки (Command Button) для действий: новый заем, сохранить, переход к списку.
- Можно добавить простой макрос/VBA: переход к другой форме, очистка полей, проверка заполненности.

Шаг 6. Тестирование в Form View

- Переключитесь в Form View.
- Протестируйте ввод данных, добавление новой записи, переключение между записями.
- Проверьте работу подформы: выбор читателя и отображение его займов (или выбор книги и указание авторов), в зависимости от выбранной структуры.

Шаг 7. Документация и представление результатов

- Сохраните форму под понятным именем, например:

- frmMemberDetailDesign (главная Members с подформой Loans)
- frmBooksWithAuthorsDesign (главная Books с подформой BookAuthors)
- Сделайте скриншоты Design View и Form View.
- Подготовьте краткий отчет: цели, структура формы, связи Master-Child, примеры сценариев использования.

5) Советы и возможные сложности

- В Design View внимательно задавайте Source и Control Source; ошибка в привязке поля приводит к пустым значениям.
- При добавлении подформы убедитесь, что связи между полями соответствуют ключам (например, MemberID, BookID).
- Если поля не редактируются, проверьте свойства Allow Edits и Data Entry, а также наличие первичных ключей.
- Для удобства используйте группировку полей и заголовки (Form Header) для понятной навигации.
- При необходимости применяйте VBA/макро для более сложной логики (проверка заполнения, автоматическое заполнение полей).

6) Формат отчёта по занятию

- Название занятия: Практическое занятие №8. Создание формы в режиме Конструктора в MS Access
- Цели и задачи
- Окружение и база данных
- Этапы создания формы: шаги и пояснения
- Результаты: сохранённые формы, скриншоты Design View и Form View
- Выводы и замечания
- Приложения: инструкции по настройке полей и связей, примеры кода (если применимо)

Практическое занятие №9 Создание отчета с помощью Мастера

1) Цели и требования

- Освоить создание типового отчета с использованием Мастера отчетов.
- Научиться выбирать источник данных (таблица/запрос), добавлять поля, настраивать группировку и сортировку.
- Изучить базовые настройки макета, заголовков, подотчетов, форматов страниц и печати.
- На примерах поработать с отчетами по займам: по читателям, по книгам и по просроченным займам.
- Подготовить краткий итог: названия отчетов, структура, используемые группировки и формат вывода.

2) Необходимое ПО и окружение

- MS Access (Office 365 / локальная версия).
- База данных из Практического занятия №1: Authors, Books, BookAuthors, Members, Loans (или эквивалентная с теми же полями).
- Резервная копия базы перед занятием.

3) Напоминание о структуре БД

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, BirthDate, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или сделайте копию.
- Убедитесь, что есть подходящие поля: для отчетов — Members (MemberID, FirstName, LastName, Email), Loans (LoanID, MemberID, BookID, LoanDate, ReturnDate, Status), Books (BookID, Title).

Шаг 1. Запуск мастера отчетов

- Вкладка Create (Создать) → Report Wizard (Мастер отчетов).
- Источник данных: выберите Loans как основную таблицу (для отчетов по займам). Можно выбрать и другой источник (например, Members или Books) в зависимости от задумки.
- Выбор полей: добавьте поля из Loans и связанные поля из Members и Books, например:
- Member: MemberID, FirstName, LastName (или FullName через вычисляемое поле в отчете).
- Book: Title (или Title через соединение).
- Показатели: LoanDate, ReturnDate, Status.

Шаг 2. Группировка и сортировка

- У Master Report Wizard можно выбрать группировку:
- Группировка 1: по читателю (Member) — вывод полного имени и сводка по его займам.
- Группировка 2: по книге (Book Title) — сводка по займам конкретной книги.
- Сортировка: по LoanDate по возрастанию или по другой логике.

Шаг 3. Макет и стиль

- Выберите макет (Columnar, Tabular, Justified) и стиль оформления.
- Wizard создаст отчет с заголовками, группами и подытогами, если вы выбрали соответствующую группировку.

Шаг 4. Завершение и базовая настройка

- Нажмите Finish. Отчет откроется в Design View или Report View (зависит от версии).
- При необходимости настройте:
- Заголовки страниц, отчета, групп.
- Подытоги (например, количество займов на группу/читателя).
- Фильтры и параметры отображения (например, показать только активные займы).

Шаг 5. Доработка в Design View

- Добавьте вычисляемые поля, если нужно:
- FullName: [FirstName] & " " & [LastName] (или через вычисляемое поле в отчете).
- DaysOverdue: DateDiff("d", [LoanDate], Nz([ReturnDate], Date())) - 14 (если просрочка считается после 14 дней).
- Настройте подотчеты (если хотите включить сводную информацию по книгам или читателям в отдельном разделе).
- Проверьте форматирование: ширина колонок, переносы строк, выравнивание, стиль шрифта.

Шаг 6. Просмотр и тестирование

- Переключитесь в Report View или Print Preview.
- Убедитесь, что данные корректно группируются, суммы/подсчеты верны, поля заполнены.
- Попробуйте экспортировать отчет в PDF или Excel (через File → Save As/Export).

Шаг 7. Документация и представление результатов

- Сохраните отчеты под понятными именами, например:
- rptMemberLoansOverview (отчет по займам читателей с группировкой по читателю)
- rptBooksLoansSummary (отчет по займам книг)
- rptOverdueLoans (отчет по просроченным займам)
- Сделайте скриншоты: окно мастера на этапе выбора полей/группировок и итоговые отчеты в Preview.
- Подготовьте краткий отчет: цели, структура отчета, группы, использованные вычисляемые поля, примеры вывода.

5) Советы и возможные сложности

- Если данные нужно фильтровать (например, показывать только активные займы или за период), добавляйте фильтр в мастере или редактируйте источник данных (Query) после завершения мастера.
- Вложенные заголовки/пояснения: используйте заголовок раздела для групп и подзаголовки в деталях.
- Вычисляемые поля в отчете: помните, что для вычислений можно использовать формулы Access, аналогичные тем, что применялись в запросах.
- При необходимости добавляйте подотчеты (subreports) для детализации внутри групп.
- Регулярно сохраняйте резервную копию базы перед созданием и правками отчетов.

6) Формат отчета по занятию

- Название занятия: Практическое занятие №9. Создание отчета с помощью Мастера в MS Access
- Цели и задачи
- Окружение и база данных
- Этапы: запуск мастера, выбор полей, группировка, макет, доработка, тестирование
- Результаты: сохраненные отчеты, скриншоты Design View и Report View
- Выводы и замечания
- Приложения: примеры списка полей и вариантов группировок, инструкции по экспорту в PDF/Excel

Практическое занятие №10 Создание отчета в режиме Конструктора

1) Цели и требования

- Освоить создание полноценных отчетов с нуля в Design View.
- Научиться вручную подбирать источник данных, размещать элементы управления и настраивать свойства.
- Освоить группировку и сортировку в отчете (Master-Detail не обязателен, но можно предусмотреть подотчет/группировки).
- На примерах поработать с отчетами по займам: по читателям, по книгам и по просроченным займам.

- Подготовить краткий итог: названия отчетов, структура, использованные вычисляемые поля и формат вывода.

2) Необходимое ПО и окружение

- MS Access (Office 365 / локальная версия).
- База данных из Практического занятия №1: Authors, Books, BookAuthors, Members, Loans (или эквивалентная с теми же полями).
- Резервная копия базы перед занятием.

3) Напоминание о структуре БД

- Authors(AuthorID PK, FirstName, LastName)
- Books(BookID PK, Title, Publisher, PublishYear)
- BookAuthors(BookID, AuthorID) — мостовая, PK: BookID + AuthorID
- Members(MemberID PK, FirstName, LastName, Email, BirthDate, ...)
- Loans(LoanID PK, BookID, MemberID, LoanDate, ReturnDate, Status)

4) Этапы занятия (пошагово)

Шаг 0. Подготовка

- Откройте базу из предыдущего занятия или создайте копию.
- Убедитесь, что для отчетов доступны поля: Members (MemberID, FirstName, LastName, Email), Loans (LoanID, MemberID, BookID, LoanDate, ReturnDate, Status), Books (BookID, Title).

Шаг 1. Создание отчета в Конструкторе (Design View)

- Вкладка Create → Design View (Конструктор).
- Источник данных: задайте Record Source у отчета (например, Loans, или объединение через запрос, если нужен доступ и к данным Members и Books).
- Разместите заголовки и пустые области для детализации.

Шаг 2. Добавление полей вручную

- В Design View используйте Field List (или добавляйте элементы управления TextBox/Label вручную).
- Добавляйте нужные поля: например, для отчета по займам — LoanDate, ReturnDate, Status, а также связанные поля из Members (FullName через выражение или отдельно FirstName/LastName) и из Books (Title).
- Привяжите каждое текстовое поле к источнику данных через свойство Control Source.
- При желании добавьте вычисляемые поля, например:
- FullName: [FirstName] & " " & [LastName]
- DaysOverdue: DateDiff("d", Nz([ReturnDate], Date()), [LoanDate]) - 14 (или другая ваша бизнес-логика).

Шаг 3. Настройка группировки и сортировки

- В Design View → Group & Sort (Группировка и Сортировка).
- Добавьте группы:
- Группа 1: по Reader (например, по MemberID или FullName) для обзора займов по читателю.
- Группа 2: по Book Title (Title) для обзора займов по книге, если нужно.
- Установите сортировку по дате займа (LoanDate) или по другой логике.
- Внесите подытоги (Count loans в группе, возможно суммы).

Шаг 4. Макет и стиль

- Выберите формат макета: Columnar, Tabular, Justified и т. д.
- Настройте заголовки отчета (Report Header), групповые заголовки, детали (Detail), подитоги (Footer).
- Отрегулируйте ширину колонок, выравнивание, переносы строк, стиль шрифта.
- При необходимости добавьте страницы, колонтитулы и номер страницы.

Шаг 5. Подотчеты и дополнительные элементы

- При необходимости добавьте Subreport (подотчеты) внутри группы для детализации (например, подотчет по авторам книги внутри займов).
- Добавьте вычисляемые поля или формулы прямо в отчете (например, возраст читателя, длительность займа).

Шаг 6. Фильтры и параметры

- В Record Source можно добавить параметры (например, показать займы за указанный период).
- Добавьте простые фильтры на уровне отчета (через свойство Filter и Filter On) или используйте параметризованный запрос как Record Source.

Шаг 7. Просмотр и тестирование

- Переключитесь в Report View или Print Preview.
- Убедитесь, что группировка корректна, данные заполняются, подитоги совпадают.
- Проверьте печать/экспорт в PDF или Excel.

Шаг 8. Документация и представление результатов

- Сохраните отчеты под понятными именами, например:
- rptMemberLoansDetailDesign (отчет по займам читателей)
- rptBooksLoansDetailDesign (отчет по займам книг)
- rptOverdueLoansDesign (отчет по просроченным займам)
- Сделайте скриншоты Design View и Report View (или Print Preview).
- Подготовьте краткий отчет: цели, структура отчета, группы и формат вывода.

5) Советы и возможные сложности

- В Design View внимательно настраивайте Record Source и каждого TextBox через свойство Control Source.
- Если данные не отображаются, проверьте свойства источника данных и наличие связанных ключей.
- При группировке убедитесь, что поля для группировки соответствуют данным (например, по MemberID или Title).
- Для вычисляемых полей используйте корректные выражения Access.
- Регулярно сохраняйте резервную копию базы перед созданием и правками отчетов.
- Если данных много, рассмотрите использование предварительного запроса (Query) как Record Source для упрощения и ускорения.

6) Формат отчета по занятию

- Название занятия: Практическое занятие №10. Создание отчета в режиме Конструктора в MS Access
- Цели и задачи
- Окружение и база данных

- Этапы: создание отчета в Design View, добавление полей, группировка, макет, доработка, тестирование
- Результаты: сохраненные отчеты, скриншоты Design View и Report View
- Выводы и замечания
- Приложения: примеры полей и вариантов группировок, инструкции по экспорту в PDF/Excel

