



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

УНИВЕРСИТЕТ (РОСБИОТЕХ)

Федеральное государственное бюджетное образовательное учреждение высшего образования
«РОССИЙСКИЙ БИОТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ (РОСБИОТЕХ)»

ФОНД ОЦЕНОЧНЫХ СРЕДСТВ ПО УЧЕБНОЙ ДИСЦИПЛИНЕ

База данных

Уровень образования:	Среднее профессиональное образование
Специальность	09.01.03 ОПЕРАТОР ИНФОРМАЦИОННЫХ СИСТЕМ И РЕСУРСОВ
Квалификация	Оператор информационных систем и ресурсов
Форма обучения	Очная
Срок освоения образовательной программы в соответствии с ФГОС (очная форма)	10 м. (на базе среднего общего образования)
Год начала подготовки	2026 г.
Период освоения дисциплины	1 семестр
Форма контроля	Контрольная работа

г. Москва 2025 г.

1. Область применения.

Фонд оценочных средств (ФОС) является неотъемлемой частью программы дисциплины при реализации программы подготовки специалистов среднего звена (ППСЗ) среднего профессионального образования (СПО) по специальности:

09.01.03 ОПЕРАТОР ИНФОРМАЦИОННЫХ СИСТЕМ И РЕСУРСОВ

Оценочные фонды разрабатываются для проведения оценки степени соответствия фактических результатов обучения при изучении дисциплины запланированным результатам обучения, соотнесенных с установленными в программе индикаторами достижения компетенций, а также сформированности компетенций, установленных программой подготовки специалистов среднего звена.

Таблица 1
Паспорт фонда оценочных средств

Компетенции	Знать:	Уметь:	Владеть навыками (иметь практический опыт):
ПК 1.4. Конвертировать аналоговые данные в цифровые	основы теории баз данных; модели данных; особенности реляционной модели и проектирование баз данных; изобразительные средства, используемые в ER-моделировании; основы реляционной алгебры; принципы проектирования баз данных; обеспечение непротиворечивости целостности данных; средства проектирования структур баз данных; язык запросов SQL	проектировать реляционную базу данных; использовать язык запросов для программного извлечения сведений из баз данных	нормализации данных; определения сущностей и связей; выбора реляционной модели; первичных ключей и внешних ключей; индексации; управления транзакциями; оптимизации запросов; резервного копирования и восстановления; безопасности данных
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			

Цели и задачи фонда оценочных средств.

Целью ФОС является установление соответствия уровня подготовки обучающихся требованиям федерального государственного образовательного стандарта ФГОС СПО по ОПОП.

ФОС предназначен для решения задач контроля достижения целей реализации ОПОП СПО и обеспечения соответствия результатов обучения области, сфере, объектам профессиональной деятельности, области знаний и типам задач профессиональной деятельности.

Задания для контрольной работы с ключами ответов

№ вопроса	Формулировка тестовых заданий	Варианты ответов	Правильный ответ
-----------	-------------------------------	------------------	------------------

1.	Что из перечисленного НЕ является функцией СУБД?	а) Управление данными во внешней памяти б) Управление транзакциями в) Создание аппаратного обеспечения сервера г) Обеспечение языков доступа к данным	в) Создание аппаратного обеспечения сервера
2.	Основным структурным элементом реляционной модели данных является:	а) Файл б) Дерево в) Таблица (отношение) г) Граф	в) Таблица (отношение)
3.	Связь "многие ко многим" (М:М) между сущностями А и В в реляционной модели реализуется с помощью:	а) Внешнего ключа в таблице А б) Внешнего ключа в таблице В в) Создания третьей (связующей) таблицы г) Не реализуется напрямую	в) Создания третьей (связующей) таблицы
4.	Первичный ключ (Primary Key) в таблице должен удовлетворять условию:	а) Может принимать пустые (NULL) значения б) Может быть составным только из числовых полей в) Быть уникальным и не равным NULL для каждой строки г) Дублироваться в связанных таблицах	в) Быть уникальным и не равным NULL для каждой строки
5.	Язык SQL, используемый для извлечения данных из базы, - это:	а) DQL (Data Query Language) б) DDL (Data Definition Language) в) DML (Data Manipulation Language) г) DCL (Data Control Language)	а) DQL (Data Query Language)
6.	Какой оператор SQL используется для добавления новых записей в таблицу?	а) UPDATE б) INSERT в) CREATE г) ADD	б) INSERT
7.	Целью нормализации баз данных является:	а) Ускорение работы базы данных за счет дублирования б) Устранение избыточности данных и аномалий обновления в) Упрощение интерфейса пользователя г) Снижение требований к аппаратному обеспечению	б) Устранение избыточности данных и аномалий обновления
8.	Сущность, атрибуты и связи отображаются на диаграмме:	а) Блок-схеме б) Диаграмме классов UML в) ER-диаграмме (Entity-Relationship) г) Диаграмме последовательности	в) ER-диаграмме (Entity-Relationship)
9.	Какой тип NoSQL базы данных оптимален для хранения иерархических, связанных данных (например, социальных сетей)?	а) Документная (Document) б) Ключ-значение (Key-Value) в) Графовая (Graph) г) Колоночная (Column)	в) Графовая (Graph)
10.	Что такое триггер в контексте баз данных?	а) Вид первичного ключа б) Хранимая процедура, автоматически выполняемая при наступлении определенного события (INSERT, UPDATE, DELETE)	б) Хранимая процедура, автоматически выполняемая при наступлении определенного

		в) Средство визуализации данных г) Тип индекса	события (INSERT, UPDATE, DELETE)
11.	Оператор SQL ALTER TABLE используется для:	а) Удаления таблицы б) Изменения структуры существующей таблицы (добавление/удаление столбцов) в) Выборки данных из таблицы г) Создания новой таблицы	б) Изменения структуры существующей таблицы (добавление/удаление столбцов)
12.	Для чего используется оператор GROUP BY в SQL-запросе?	а) Для сортировки результатов б) Для фильтрации строк в) Для группировки строк, имеющих одинаковые значения в указанных столбцах г) Для ограничения количества выводимых строк	в) Для группировки строк, имеющих одинаковые значения в указанных столбцах
13.	Какая агрегатная функция SQL вычисляет среднее значение?	а) COUNT() б) SUM() в) AVG() г) MAX()	в) AVG()
14.	Процедура резервного копирования БД необходима в первую очередь для:	а) Ускорения работы системы б) Восстановления данных после сбоев или удаления в) Увеличения объема хранимых данных г) Шифрования информации	б) Восстановления данных после сбоев или удаления
15.	Индекс в базе данных создается преимущественно для:	а) Ускорения операций поиска и выборки данных б) Увеличения занимаемого места на диске в) Обеспечения ссылочной целостности г) Резервирования данных	а) Ускорения операций поиска и выборки данных
16.	Установите соответствие между этапом проектирования БД и его основным результатом.	Зоны (Этапы): А. Сбор и анализ требований Б. Концептуальное проектирование В. Логическое проектирование Г. Физическое проектирование Реквизиты (Результаты): 1. ER-диаграмма 2. Техническое задание, список бизнес-требований 3. Набор таблиц со связями, нормализованная схема БД 4. Скрипты создания БД, индексы, параметры хранения	1 — Б 2 — А 3 — В 4 — Г
17.	Установите соответствие между типом SQL-запроса и его описанием.	Зоны (Типы запросов): А. DDL (Data Definition Language) Б. DML (Data Manipulation Language) В. DQL (Data Query Language) Г. DCL (Data Control Language) Реквизиты (Описания):	1 — В 2 — Г 3 — А 4 — Б

		<ol style="list-style-type: none"> 1. SELECT — выборка данных 2. GRANT, REVOKE — управление правами доступа 3. CREATE, ALTER, DROP — работа со структурой объектов БД 4. INSERT, UPDATE, DELETE — изменение данных 	
18.	Установите соответствие между понятием в проектировании БД и его определением.	<p>Зоны (Понятия):</p> <p>А. Первичный ключ (Primary Key) Б. Внешний ключ (Foreign Key) В. Нормализация Г. Индекс</p> <p>Реквизиты (Определения):</p> <ol style="list-style-type: none"> 1. Процесс устранения избыточности и аномалий данных 2. Поле для связи с первичным ключом другой таблицы 3. Уникальный идентификатор записи в таблице 4. Структура для ускорения поиска данных 	<p>1 — В 2 — Б 3 — А 4 — Г</p>
19.	Установите соответствие между типом связи в ER-диаграмме и её реализацией в реляционной модели.	<p>Зоны (Типы связей):</p> <p>А. Один к одному (1:1) Б. Один ко многим (1:M) В. Многие ко многим (M:M)</p> <p>Реквизиты (Способ реализации):</p> <ol style="list-style-type: none"> 1. Добавление внешнего ключа в одну из таблиц 2. Создание промежуточной таблицы с двумя внешними ключами 3. Объединение сущностей в одну таблицу или размещение внешнего ключа в любой из таблиц 	<p>1 — Б 2 — В 3 — А</p>
20.	Установите соответствие между оператором SQL и его назначением.	<p>Зоны (Операторы):</p> <p>А. WHERE Б. GROUP BY В. ORDER BY Г. HAVING</p> <p>Реквизиты (Назначение):</p> <ol style="list-style-type: none"> 1. Сортировка результатов запроса 2. Фильтрация данных после группировки 3. Группировка строк для агрегатных функций 4. Фильтрация строк по 	<p>1 — В 2 — Г 3 — Б 4 — А</p>

		условию	
21.	Перечислите основные этапы проектирования базы данных.		Сбор и анализ требований. 2. Концептуальное проектирование (построение ER-диаграммы). 3. Логическое проектирование (преобразование в реляционную схему, нормализация). 4. Физическое проектирование (выбор СУБД, определение индексов, типов данных). 5. Реализация и тестирование. 6. Эксплуатация и сопровождение.
22.	Что такое JOIN в SQL? Назовите и кратко опишите два основных типа JOIN.		JOIN — оператор для объединения строк из двух или более таблиц на основе связанного столбца. Основные типы: 1) INNER JOIN — возвращает только те строки, где есть соответствие в обеих таблицах. 2) LEFT (OUTER) JOIN — возвращает все строки из левой таблицы и соответствующие строки из правой; если соответствия нет, возвращается NULL для полей правой таблицы.
23.	Объясните, чем отличается уникальный ключ (UNIQUE) от первичного ключа (PRIMARY KEY).		Оба обеспечивают уникальность значений в столбце(ах). Отличия: 1) В таблице может быть только один PRIMARY KEY, но несколько UNIQUE-ограничений.

			<p>2) PRIMARY KEY не допускает NULL-значений, а UNIQUE — допускает (одно NULL-значение на столбец, в зависимости от СУБД).</p> <p>3) PRIMARY KEY часто используется для организации связей между таблицами.</p>
24	Для чего нужен оператор HAVING в SQL и чем его использование отличается от WHERE?		<p>HAVING используется для фильтрации результатов группировки (GROUP BY) по условию, применяемому к агрегатным функциям (например, COUNT(...) > 5). WHERE же фильтрует отдельные строки перед группировкой и не может работать с агрегатными функциями.</p>
25.	Назовите не менее трех мер (инструментов) обеспечения безопасности данных в СУБД.		<p>1) Система разграничения прав доступа (роли, пользователи, привилегии: GRANT/REVOKE). 2) Аутентификация (проверка подлинности пользователя). 3) Резервное копирование и восстановление. 4) Аудит (ведение журналов действий). 5) Шифрование данных.</p>

Оценивание обучающегося на занятиях осуществляется в соответствии с локальным актом университета (положением), регламентирующим проведение текущего контроля успеваемости и промежуточной аттестации обучающихся и организации учебного процесса с применением балльно-рейтинговой системы оценки качества обучения.

Промежуточная аттестация по дисциплине

Форма промежуточной аттестации – .Контрольная работа

Контрольная работа проводится с целью проверки уровня усвоения теоретических знаний и практических умений по дисциплине, а также степени сформированности профессиональных компетенций (ПК), указанных в рабочей программе.

2. Структура и содержание контрольной работы

Контрольная работа включает 25 заданий, разделенных на три части:

- Часть 1: 15 тестовых заданий с выбором одного правильного ответа.
- Часть 2: 5 заданий на установление соответствия.
- Часть 3: 5 открытых вопросов, требующих развернутого ответа.

3. Критерии и баллы за выполнение заданий

№ части	Тип задания	Кол-во	Макс. балл за задание	Макс. балл за часть	Критерии оценивания
1	Тест с выбором ответа	15	1 балл	15 баллов	1 балл начисляется за каждый правильно выбранный вариант ответа. За неправильный или отсутствующий ответ – 0 баллов.
2	Установление соответствия	5	2 балла	10 баллов	За каждое полностью верное соответствие (все связи установлены правильно) – 2 балла. За частично верное соответствие – 1 балл (например, 2-3 правильные связи из 4). За неверное – 0 баллов.
3	Открытый вопрос (развернутый ответ)	5	3 балла	15 баллов	3 балла: Полный, развернутый, точный ответ, соответствующий ключу. Логичное изложение, использование профессиональной терминологии. 2 балла: Ответ верен по сути, но неполон, содержит незначительные неточности или недостаточно развернут. 1 балл: Ответ поверхностный, содержит существенные ошибки или раскрывает лишь часть вопроса.

					0 баллов: Ответ неверен, отсутствует или не соответствует вопросу.
--	--	--	--	--	--

Максимально возможная сумма баллов за контрольную работу: $15 + 10 + 15 = 40$ баллов.

4. Шкала перевода баллов в оценку (пятибалльная система)

Итоговая оценка за контрольную работу выставляется по следующей шкале:

Сумма набранных баллов (из 40)	Оценка по 5-балльной системе	Качественная характеристика
36 – 40 баллов	5 (Отлично)	Обучающийся демонстрирует глубокие и системные знания. Ответы точные, полные, логически выстроенные. Практические задания выполнены безошибочно.
28 – 35 баллов	4 (Хорошо)	Обучающийся показывает твердые знания материала. Допускаются незначительные неточности или неполнота в 1-2 ответах. Основные понятия усвоены.
20 – 27 баллов	3 (Удовлетворительно)	Обучающийся обнаруживает минимально достаточные знания. Ответы фрагментарны, содержат ошибки, но ключевые вопросы раскрыты. Пороговый уровень пройден.
Менее 20 баллов	2 (Неудовлетворительно)	Обучающийся не усвоил основное содержание дисциплины. Ответы неверные, отсутствуют или не соответствуют вопросу. Требуется повторное изучение темы.

Пример практических работ

Лабораторная работа 1

-

Цель: Изучение типов логических моделей данных

Задание

1. На основании лекционного материала подготовить схемы баз данных разных логических моделей – реляционной, иерархической, сетевой. Темы баз данных:
корабли второй мировой
автосервис

*компьютерная фирма
аэропорт
регистратура поликлиники
журнал колледжа
склад готовой продукции фирмы вторсырья
анкеты студентов
магазин автозапчастей*

2. В реляционной модели данных должны быть связи всех типов – один к одному, один ко многим и многие ко многим.

Лабораторная работа 2

Цель: Изучение способов организации памяти хранения данных и типов данных в таблицах

Задание

1. Выбрать предметную область (например: склад, больница, аптека, аэропорт и т.д.) и составить для нее:

а) описание предметной области (от имени конечного пользователя);

б) ER-диаграмму.

Ограничение: от 5ти до 10ти сущностей для описания области. Каждый объект должен иметь хотя бы один атрибут.

Описание и диаграмма включается в отчет по лабораторной работе. Отчет выполняется в печатном виде на листах формата А4 согласно общепринятым правилам оформления лабораторных работ.

2. Используя MS Access перенести полученную модель в БД, используя таблицы и схему данных.

Лабораторная работа 3

Цель: научиться использовать операторы языка SQL для работы с данными БД.

Задание

1. Написать запрос для выбора автомобилей определенного цвета. Цвет задается в виде параметра в условии WHERE (например, 'белый').
2. Определить количество автомобилей, у которых номер фондового извещения начинается на "10" и не заканчивается на "39"
3. По каждой штатной группе а/м определить, сколько а/м каждой марки было выпущено в заданном году. Вывести названия групп и названия марок на экран.
4. Определить, какие а/м данного класса переданы в подразделения после указанной даты. Указать также номер автомобиля и дату документа передачи каждого а/м.
5. Произвести выборку автомобилей из двух полей «номер авто», «класс авто» (подставлять название из отношения MENU). Если поле «класс» в таблице MENU не существует, то выводить строку «Класс средства неизвестен» с помощью функции iif.
6. Определить, сколько а/м каждой марки имеют год выпуска меньший, чем округленный до целого средний год выпуска а/м заданной пользователем марки.
7. Определить какое количество а/м каждой марки в каком году было произведено (перекрестный запрос: марки а/м на год производства)

Лабораторная работа 3

Цель: Развитие у студентов навыков программирования приложений, использующих БД, знакомство с частями SDL и DML языка SQL.

Задание

1. Создать базу данных по любой предметной области (желательно по курсовой работе), которая должна минимум содержать таблицу, состоящую минимум из 6 полей, где обязательно должно присутствовать поле типа date. Для создания таблиц БД использовать скриптовый файл или макрокоманду, содержащую набор SQL-команд из части языка SDL;
 2. Реализовать процедуры Добавления, Удаления, Поиска и Изменения, с помощью SQL;
 3. Организовать оконный интерфейс для функций, созданных на предыдущем этапе (добавления, удаления, поиска и изменения);
 4. Поиск должен осуществляться с использованием индексов, т.е. поля, по которым осуществляется поиск, должны быть проиндексированы. Для создания индексов использовать CREATE INDEX.
- Выполнение обязательных пунктов = 70%
Бонус (+ 15%): Для получения дополнительных баллов реализовать кодовые поля в основной таблице и справочник(и) для расшифровки этих полей (подобно базе allauto.mdb).

Бонус (+ 15%): Организовать механизм авторизации – вход в БД по паролю для нескольких пользователей (статья справки «Пароли (MDB)»).

Лабораторная работа 4

Цель: моделирования механизмов расчетов в запросах, протоколирование и отката команд.

Задание

1. Необходимо реализовать главную форму, запускаемую автоматически при открытии БД (для allauto.mdb). Эта форма должна позволять редактировать данные о а/м в таблице AUTO: добавление, удаление, изменение автомобиля (или автомобилей). При запуске приложения в нормальном режиме не выводить окно базы данных.

2. Реализовать протоколирование – журнал изменений. Должны быть реализованы функции отката изменений БД (таких как добавление, удаление, изменение записи). Для этого организовать специальную форму, позволяющую осуществлять:

- "Откат назад" – откат назад на одно изменение в базе (не активен, если не было изменений или выполнены все откаты назад);
- "Откат вперед" – откат вперед на одно изменение (может быть не активен). Не забудьте проиндексировать таблицу MENU.

Выполнение обязательных пунктов = 55%

Бонус (+ 15%): предусмотреть, что при аварийном завершении программы, существует возможность восстановления всей цепочки отката, и лишь только при "нормальном" завершении работы с программой цепочка отката обнуляется (для этого используйте макросы).

Бонус (+ 30%): Реализовать процедура поиска для главной формы. "Поиск" - содержит следующие подпункты (функции):

1. "Параметры" - выдается форма, содержащая перечень не менее 3х кодовых полей (любые, из таблицы AUTO). При выборе любого из этих пунктов выдается перечень возможных значений, который может принять данное кодовое поле (выпадающий список значений из menu). У значений могут быть свои подзначения, поэтому необходимо организовать рекурсивную (или циклическую) процедуру обхода дерева значений (словаря данных) – т.е. для каждого поля выводить не только его значения, но и подзначения. У ВСЕХ ДОЛЖНО БЫТЬ КОДОВОЕ ПОЛЕ "МАРКА ТРАНСПОРТНОГО СРЕДСТВА".

При выборе значений, заносить их в строку поиска, типа: <код. поле>=<значение> AND

2. "Строка поиска" - выводит на экран содержимое строки поиска.

3. "Очистить строку поиска" - обнуляет строку поиска.

4. "Найти" – поиск записей в таблице auto в соответствии со строкой поиска, если строка поиска

пуста, то выводятся все записи.

Все поля, которые используются для поиска, должны быть расшифрованы.

Лабораторная работа 5

Цель: научиться создавать удобные экранные формы для трёх основных видов работ в информационной системе: просмотра, ввода и редактирования, научиться пользоваться этими формами, изучить различные права доступа к данным.

Задание

Пользователям информационных систем предоставляются разные права доступа к данным. Одним разрешается только просмотр, другим – просмотр и редактирование и т. д. Для удобства работы с данными и обеспечения прав собственности разрабатывают три вида форм: для *просмотра, ввода и редактирования*.

В формах для просмотра запрещено изменять данные, поэтому в них не нужно отображать списки допустимых значений. Так как при просмотре желательно видеть сразу много записей, то формы этого вида открывают в виде таблиц. Форму для редактирования удобно открывать сразу на записи, в которую нужно внести изменение, и отображать в ней списки допустимых значений.

Форма для ввода в момент начала ввода новой записи содержит пустые поля. Если набор вариантов вводимых данных небольшой, то следует пользоваться списками допустимых значений.

Создание формы для просмотра. В окне базы данных выберите объект «Формы» и пункт меню «Создать». Откроется окно «Новая форма». Выберите в верхней части окна пункт меню «Конструктор» и в нижней части окна укажите таблицу «преподаватели». Нажмите «ОК». Должны появиться 3 окна: форма, список полей и панель элементов. Список полей и панель элементов можно открыть через пункт меню Access «Вид» в верхней части экрана.

Перетащите мышкой из списка полей в форму поля «ФИО», «Дата_рожд.», «Зарплата», «Биография» и «Фотография». Эти поля таблицы отобразятся в форме также в виде полей. Для поля «Фотография» будет создан элемент «Присоединённая рамка объекта».

При перетаскивании в форму поля «Должность» Access создаёт элемент «список», в котором будут показаны все допустимые значения должности. Так как при просмотре список не нужен, то для «Должности» нужно создать элемент «поле» вручную. Для этого щёлкните мышкой по элементу «поле» на панели элементов. Переместите указатель мышки в форму и раздвиньте поле до подходящих, на Ваш взгляд, размеров. Введите вместо «поле0» название поля. Теперь нужно привязать поле формы к полю «Должность» таблицы «преподаватели». Щёлкните правой кнопкой мышки по создаваемому полю в форме. В появившемся меню выберите пункт «Свойства». Появится окно свойств поля. Выберите вкладку «Данные» и в свойстве «Данные» выберите «Должность».

Аналогично создайте поля для «Степени» и «Звания».

Для ввода названия формы откройте раздел формы «Заголовок», выполнив пункты меню Access «Вид» ® «Заголовок/примечание формы». Название формы должно выглядеть примерно так:

ПРЕПОДАВАТЕЛИ

(форма для просмотра)

В форме, предназначенной только для просмотра, необходимо запретить пользователю вносить какие-либо изменения в данные. Щёлкните правой кнопкой мышки по квадратику в левом верхнем углу формы и в раскрывшемся меню выберите пункт «Свойства». Должно открыться окно «Форма». Выберите вкладку «Данные». Установите для свойства «Тип набора записей» значение «Статический набор».

Отредактируйте по своему вкусу названия полей и формы.

Перейдите в режим формы и проверьте её работу. Форму можно просматривать в трёх режимах:

- простая форма,
- ленточная форма,
- таблица.

Чаще всего при просмотре используется режим «Таблица». Просмотрите форму во всех трёх режимах. Режимы просмотра устанавливаются Конструктором. В режиме конструктора откройте окно свойств формы и выберите вкладку «Макет». Подберите нужные значения свойств «Режим по умолчанию» и «Допустимый режим». Режим «Ленточная форма» не может использоваться при включении в форму подчинённых форм.

Создание формы для редактирования. Воспользуйтесь помощью мастера. В окне базы данных выберите объект «Формы» и затем – «Создание формы с помощью мастера». В появившемся окне «Создание формы» выберите таблицу «преподаватели» и все поля за исключением поля «код_преп». Следуйте далее указаниям мастера. Форму назовите «преп_ред». Если при просмотре Вам не понравится вид формы, то перейдите в режим конструктора и внесите изменения в свойства либо всей формы, либо отдельных элементов.

Для проверки работы формы внесите изменения в данные о преподавателях.

Создание формы для ввода новых записей. Форма для ввода отличается от формы для редактирования только значением свойства «Ввод данных».

Скопируйте форму «преп_ред». Для этого в окне базы данных щёлкните правой клавишей мышки по названию формы «преп_ред» и выберите «копировать», затем на любом свободном месте окна базы данных снова щёлкните правой клавишей мышки и выберите «вставить». Назовите форму «преп_ввод». Откройте форму «преп_ввод» в режиме конструктора и измените значение свойства «Ввод данных» на «да». Значение «да» переводит форму в режим, в котором не видны уже имеющиеся в таблице записи, все поля формы пустые. После ввода данных очередной записи и перехода к следующей введённая запись автоматически заносится в базу и форма готова к вводу новой записи.

Удобно заполнять поля, когда они расположены друг под другом. Перетащите мышкой поля и их названия так, чтобы каждое поле со своим названием (кроме фотографии) располагалось в отдельной строке. Фотографию можно расположить справа от остальных полей.

Перейдите в режим формы и введите данные о двух-трёх преподавателях.

Лабораторная работа 6

Цель: научиться создавать форму, с помощью которой множество объектов (формы, отчёты, запросы) объединяются в единую, управляемую пользователем систему с соответствующими атрибутами и ключами.

-

Задание

Вся работа пользователя с информационной системой организуется через главную, как правило, кнопочную форму. Создадим форму (рис. 1), из которой будут вызываться созданные ранее формы.

Создание формы, не связанной с таблицей базы данных. Создайте форму в режиме конструктора, не указывая источник данных.

Рис. 1. Пример формы

Создание поля со списком для выбора записи. Создайте поле со списком преподавателей. Из этого списка пользователь сможет выбрать преподавателя, сведения о котором нужно просмотреть. Выберите на панели элементов поле со списком и поместите его в форму. Подтвердите, что поле со списком будет использовать значения из таблицы или запроса и щёлкните по кнопке «далее». Выберите таблицу «преподаватели» и снова щёлкните по кнопке «далее».

Для отображения в поле со списком нужно кроме поля «ФИО» выбрать ключевое поле «код_преп» и показывать в форме при просмотре списка только «ФИО». Все значения ключевого поля по определению оригинальны. Люди с одинаковыми фамилиями и инициалами иногда встречаются. Если указать для отображения в списке только поле «ФИО», то Access (и другие СУБД) всегда будет выбирать из таблицы первую по порядку из двух записей, имеющих одинаковые значения поля «ФИО». Если в строке списка есть и «код_преп» и «ФИО», то из таблицы будет выбрана запись, содержащая значение поля «код_преп».

Для быстрого выбора из длинного списка можно в поле ввести первую букву нужной фамилии.

Создание кнопки для просмотра данных обо всех преподавателях. Если нужны данные о многих преподавателях, то целесообразно открыть форму для просмотра в режиме таблицы. Создадим кнопку и связанную с ней процедуру, открывающую форму в режиме таблицы.

Выберите на панели элементов кнопку и поместите её в форму. В открывшемся окне «Создание кнопок» выберите категорию «Работа с формой» и действия «Открыть форму». Далее выберите форму «преп_просм», далее – переключатель «Открыть форму и показать все записи». далее – переключатель «Текст» и наберите в поле надпись на кнопке «Просмотр всех преподавателей». Задайте имя кнопки «откр_таб_преп». Созданная кнопка будет открывать «преп_просм» в режиме формы.

Для того чтобы форма «преп_просм» открывалась в режиме таблицы, внесите изменения в процедуру, которую Access автоматически создал вместе с кнопкой. Для этого в режиме конструктора откройте окно свойств кнопки «откр_таб_преп» и в нём вкладку «События». Щёлкните мышкой по свойству «Нажатие мышки», а затем щёлкните по квадратной кнопке с многоточием, расположенной справа. Откроется окно Visual Basic. В процедуре «Откр_таб_преп_Click()» в строчку

```
DoCmd.OpenForm stDocName, , , stLinkCriteria
```

вставьте после первой запятой слово «acFormDS» (это параметр, задающий открытие формы в режиме таблицы). Строка примет вид:

```
DoCmd.OpenForm stDocName, acFormDS, , stLinkCriteria
```

Эта строка содержит команду на открытие формы. Значения «слов» в строке следующие:

- **DoCmd** – выполнить команду;
- **OpenForm** – команда «открыть форму»;
- **StDocName** – переменная, содержащая имя формы.

Через запятую перечислены параметры команды.

Закройте окно Visual Basic и проверьте работу кнопки.

Создание кнопок для просмотра и редактирования данных об определённом преподавателе.

Создадим кнопку, с помощью которой будет открываться форма для просмотра данных об одном преподавателе. ФИО преподавателя выбирается из поля со списком.

Выберите на панели элементов кнопку и поместите её в форму. В открывшемся окне «Создание кнопок» выберите категорию «Работа с формой» и действия «Открыть форму». Далее выберите форму «преп_просм», далее – переключатель «Открыть форму для отобранных записей».

В следующем окне нужно указать, что в открываемой форме «преп_просм» будет показана запись из таблицы «преподаватели», содержащая значение поля «код_преп» (поля из таблицы базы данных), равное выбранному из поля со списком в форме «кафедра».

Далее выберите переключатель «Текст» и наберите в поле надпись на кнопке «Просмотр одного преподавателя». Задайте имя кнопки «откр_один_преп».

Кнопка для редактирования данных о преподавателе создаётся аналогично. Отличие состоит только в названиях кнопки и открываемой формы. Назовите кнопку «Редактирование и удаление», так как в дальнейшем в форме для редактирования будет кнопка для удаления записи.

Кнопка для ввода записи создаётся аналогично двум предыдущим, только не нужно связывать поля и исправлять процедуру. Выберите переключатель «Открыть форму и показать все записи», надпись на кнопке «Ввод преподавателя». Задайте имя кнопки «ввод_преп».

Кнопка для удаления записи. Чтобы уменьшить возможности ошибиться при удалении записи, необходимо:

- в момент нажатия кнопки удаления видеть удаляемую запись,
- после нажатия кнопки для удаления записи пользователь должен подтвердить приказ об удалении.

Создайте кнопку для удаления записи в форме для редактирования. Выберите категорию «Обработка записей» и действие «Удалить запись». На кнопке напишите «Удаление». Попробуйте удалить одну запись.

Внимание! При вводе новых записей номер удалённой записи не используется. В последовательности номеров образуется неустрашимый разрыв.

На рис. 1 в форме «кафедра» изображена необязательная кнопка «Удаление». Для её работы используется следующая процедура:

```
Private Sub КнопкаУд_Click()

    Dim N_record As String

    Dim dbs As Database

    a = MsgBox("Удалять?", vbYesNo)

    If a = vbNo Then

        Exit Sub

    Else

        N_record = Me![ПолеСоСписком0]

        Set dbs = CurrentDb

        dbs.Execute "DELETE FROM преподаватели WHERE код_преп=" & N_record

    End If

End Sub
```

Процедурой *КнопкаУд_Click()* удаляется из таблицы преподаватели запись, код (поле «код_преп») которой равен значению кода преподавателя, выбранному из поля со списком в форме «кафедра» (Me![ПолеСоСписком0]). Следует отметить, что перед удалением записи пользователь должен видеть всё её содержимое. Поэтому лучше кнопку для удаления поместить только в форме для просмотра или в форме для редактирования.

Самостоятельно создайте кнопки для выхода из форм для просмотра, редактирования и ввода.

Лабораторная работа 2

Лабораторная работа

Цель: научиться устанавливать связи между таблицами и строить подчинённые формы.

Задание

Цель работы: научиться устанавливать связи между таблицами и строить подчинённые формы.

Между реальными объектами, данные о которых хранятся в базе, могут существовать логические связи. Например, многие группы студентов входят в состав одного факультета, и ни одна группа не может относиться сразу к двум факультетам. Говорят, что между факультетом и группой имеется связь «один ко многим».

Отображение связи «один ко многим». В режиме конструктора дополните структуру таблицы «Группы» полем «код_ф», затем перейдите в режим просмотра таблицы и распределите группы по факультетам, то есть введите в поле «код_ф» коды факультетов, хранящиеся в поле «код_фак» таблицы «факультет».

Подмена кода объекта именем. Access позволяет во время просмотра данных в таблице «Группы» подменять код факультета его названием, взятым из таблицы «факультет». Для этого откройте таблицу «Группы» в режиме конструктора, выберите в свойствах поля «код_ф» вкладку «Подстановка» и установите следующие значения свойств:

- для свойства «Тип элемента управления» - Поле со списком,
- для свойства «Тип источника строк» - Таблица или запрос,
- для свойства «Источник строк» - факультет,
- для свойства «Присоединённый столбец» - 1,
- для свойства «Число столбцов» - 2,
- для свойства «Ширина столбцов» – 0,
- для свойства «Ограничиться списком» – да.

Теперь перейдите в режим таблицы. В поле «код_ф» должны появиться названия факультетов.

Внесение связей между таблицами в схему данных. В Access можно создать схему данных, в которой явно указываются связи между полями разных таблиц. Схема данных используется для поддержки целостности базы данных и, кроме того, упрощает построение форм и запросов.

Закройте таблицу «Группы». Выполните пункты меню Сервис ® Схема данных. В любом месте открывшегося окна щёлкните правой кнопкой мышки. В появившемся меню выберите пункт «Добавить таблицу» и добавьте таблицы «Группы» и «факультет».левой кнопкой мышки соедините поле «код_фак» и поле «код_ф» связываемых таблиц. Должно появиться окно «Изменение связей».

Установите флажок «Обеспечение целостности данных». Теперь, если не устанавливать два других флажка, Access при попытке удалить из таблицы «факультет» любую запись будет проверять, нет ли в таблице «Группы» кода удаляемой записи. Если такой код есть, то появится сообщение об ошибке. Другими словами, пока в базе данных указано, что на факультете есть хотя бы одна группа, удалить этот факультет нельзя.

Флажок «Каскадное удаление связанных записей» служит для автоматического удаления из таблицы «Группы» всех записей, связанных с записью, удаляемой из таблицы «факультет».

Флажок «Каскадное обновление связанных полей» служит для автоматического обновления в таблицы со стороны «многие» значения связанного поля, обновляемого в таблице со стороны «1». В нашем случае связанное поле – счётчик и оно не может обновляться.

После закрытия окна «Изменение связей» на схема данных появится связь «один ко многим» между таблицами «факультет» и «Группы» (рис. 2).

Создание подчинённой формы. Создайте новую форму. Назовите её «факультеты». Отобразите в ней все поля, кроме «код_фак».



Рис. 2. Схема данных

Теперь в форме «факультеты» нужно создать подчинённую форму, в которой будут показываться данные о группах, входящих в состав показываемого в главной форме факультета. Раздвиньте границы формы «факультеты» так, чтобы в ней поместилась таблица с данными о группах. На панели элементов найдите элемент «Подчинённая форма/отчёт» и установите его в форме «факультеты».

Скриншот формы «факультет» с подчинённой формой «Группы». В верхней части формы «факультет» расположены поля: факультет (Математика), телефон (246), декан (Петров Н.Н.), секретарь (Петрова М.Н.). В центре находится таблица с данными о группах:

Ном_групп	Специальность	Курс
124	теория вероятностей	1
217	теория вероятностей	2
321	теория вероятностей	3
451	анализ	2

В нижней части формы «факультет» расположены кнопки для навигации по записям: «Запись: 3 из 4».

Рис.3. Форма *факультет* с подчинённой формой

В окне мастера подчинённых форм установите переключатель «Имеющиеся таблицы и запросы» и щёлкните по кнопке «Далее». В следующем окне мастера подчинённых форм выберите таблицу «группы» и все поля, кроме «Код_гр». Выберите переключатель «Выбор из списка» и оставьте предложенное название подчинённой формы. На рис. 3 показана форма «факультет» с подчинённой формой, в которой выведены группы, входящие в состав просматриваемого в основной форме факультета. Перейдите в режим формы и просмотрите несколько записей.

