

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Худин Александр Николаевич

Должность: Ректор

Дата подписания: 11.04.2016 13:17:41

Уникальный программный ключ:

08303ad8de1c60b987361de7085acb509ac3da143f415362ffaf0ee37e73fa19

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Курский государственный университет»

Колледж коммерции, технологий и сервиса

Методические рекомендации
по выполнению практических работ
по дисциплине:
«БАЗЫ ДАННЫХ»

для студентов 2 курса специальности
09.02.05 Прикладная информатика (по отраслям)



Составитель: преподаватель
Негребецкая В.И.

Курск 2017

СОДЕРЖАНИЕ

Практическая работа № 1	
Тема: Классификация СУБД. Создание простых баз данных	3
Практическая работа № 2	
Тема: Построение семантических моделей	7
Практическая работа № 3	
Тема: Построение инфологических моделей	13
Практическая работа № 4	
Тема: Реляционная модель данных	15
Практическая работа № 5	
Тема: Преобразование инфологической модели данных в реляционную	18
Практическая работа № 6	
Тема: Язык запросов SQL.....	21
Практическая работа № 7	
Тема: Язык запросов SQL.....	26
Практическая работа № 8	
Тема: Проектирование баз данных.....	28
Практическая работа № 9	
Тема: Создание таблиц, ввод и редактирование данных в MS Office Access 2007	31
Практическая работа № 10	
Тема: Поиск данных в MS Office Access 2007	37
Практическая работа № 11	
Тема: Создание и использование форм для ввода и редактирования данных в MS Office Access 2007	41
Практическая работа № 12	
Тема: Создание отчетов в MS Office Access 2007	52
Практическая работа № 13	
Тема: Реляционные базы данных в MS Office Access 2007	57
Практическая работа № 14	
Тема: Создание запросов в Access с помощью SQL 2007	65
Практическая работа № 15	
Тема: Разработка приложений в MS Office Access 2007.....	76
Список рекомендованной литературы.....	88

Практическая работа № 1

Тема: Классификация СУБД. Создание простых баз данных

Цель: углубление теоретических знаний о СУБД; изучение и освоение основных принципов классификации существующих систем управления баз данных; сформировать навыки по созданию БД, таблиц.

Теоретический материал:

Классификация СУБД

По модели данных

По типу управляемой базы данных СУБД разделяются на:

- иерархические
- сетевые
- реляционные
- объектно-реляционные
- объектно-ориентированные.

По архитектуре организации хранения данных:

- локальные СУБД (все части локальной СУБД размещаются на одном компьютере).
- распределенные СУБД (части СУБД могут размещаться на двух и более компьютерах).

По способу доступа к БД

- Файл-серверные.

В файл-серверных СУБД файлы данных располагаются централизованно на файл-сервере. Ядро СУБД располагается на каждом клиентском компьютере. Доступ к данным осуществляется через локальную сеть. Синхронизация чтений и обновлений осуществляется посредством файловых блокировок. Преимуществом этой архитектуры является низкая нагрузка на ЦП сервера, а недостатком — высокая загрузка локальной сети.

На данный момент файл-серверные СУБД считаются устаревшими.

Примеры: Microsoft Access, Borland Paradox.

- Клиент-серверные.

Такие СУБД состоят из клиентской части (которая входит в состав прикладной программы) и сервера (см. *Клиент-сервер*). Клиент-серверные СУБД, в отличие от файл-серверных, обеспечивают разграничение доступа между пользователями и мало загружают сеть и клиентские машины. Сервер является внешней по отношению к клиенту программой, и по надобности его можно заменить другим. Недостаток клиент-серверных СУБД в самом факте существования сервера (что плохо для локальных программ — в них удобнее встраиваемые СУБД) и больших вычислительных ресурсах, потребляемых сервером.

Примеры: Firebird, Interbase, IBM DB2, MS SQL Server, Sybase, Oracle, PostgreSQL, MySQL, ЛИНТЕР.

- Встраиваемые.

Встраиваемая СУБД — библиотека, которая позволяет унифицированным образом хранить большие объёмы данных на локальной машине. Доступ к данным может происходить через SQL либо через особые функции СУБД. Встраиваемые СУБД быстрее обычных клиент-серверных и не требуют установки сервера, поэтому востребованы в локальном ПО, которое имеет дело с большими объёмами данных (например, геоинформационные системы).

Примеры: OpenEdge, SQLite, BerkeleyDB, один из вариантов Firebird, один из вариантов MySQL, Sav Zigzag, Microsoft SQL Server Compact, ЛИНТЕР.

Задания:

1. Составить схему «Классификация СУБД» с описанием характеристики современных СУБД и примерами.
2. Установить Firebird.
 - раскрыть двойным щелчком лкм файл Firebird-2.5.1.26351_1_Win32;
 - в окне установки последовательно выбрать, используя кнопку Далее:
 - Выполнить;
 - Русский;
 - Я принимаю условия соглашения;
 - Снять активацию с позиции After installation What next?
3. Открыть панель управления (Пуск – Настройки – панель управления) и проверить наличие Firebird Server Manager. Открыть значок, проверить наличие автоматического запуска в качестве сервера.
4. Установить IBExpert.
 - раскрыть двойным щелчком лкм файл setup_trial;
 - в окне установки последовательно выбрать:
 - Выполнить;
 - Next;
 - Я принимаю условия соглашения;
 - Снять активацию со всех позиций, кроме установленного по умолчанию IBExpert ID;
 - Снять активацию с позиции Create a Quick Launch icon (создание иконки на рабочем столе);
 - Finish.
5. Открыть с рабочего стола IBExpert.
6. Перевести базу данных на русский язык (Настройки – Настройки среды – Язык интерфейса).
7. Создать папку «Project».
8. В ней создать базу данных «Проект». При создании БД использовать: локальный сервер, файл базы данных (местоположение): C:\Проект, имя пользователя SYSDBA, пароль – masterkey, кодировка – Win1251.
9. В следующем окне выбрать: Сервер - Firebird 2.5, описание – Проект.
10. В левой части щелкнуть два раза лкм по Проект.

11. Создать таблицу «List_phones» (в правой части окна щелкнуть лкм по Таблицы, выбрать Новая таблица ...).

12. Поле счетчик (ID), активизировать положение «Не пустой» и «AutoInc». В последней позиции щелкнуть два раза лкм и выбрать «Создать генератор», «Создать триггер».

13. Создать следующие поля:

- surname (Фамилия) – Varchar, длина 30
- fname (Имя) - Varchar, длина 30
- patronymic (Отчество) - Integer
- tel_num (телефонный номер) – Integer

14. Нажать на панели инструментов кнопку «Компиляция».

15. Изменить поле Patronymic, задав значение Varchar, для этого на поле щелкнуть 2 раза лкм, в появившемся окне выбрать «Редактировать домен», задать значение Varchar, длина 30.

16. Создать Новую процедуру Proc_List_Phones (лкм на поле Процедура).

17. В правой нижней части окна оставить строчки begin и end.

18. В правой верхней части окна создать входные параметры:

Имена полей – fin_id, fin_surname (фамилия), fin_fname (имя), fin_patronymic (Отчество), fin_tel_num (Телефон).

19. Типы и длину полей задать такие же, как и в таблице.

20. В правой нижней части вписать следующий код программы (после begin):

```
If (:fin_id < 0) then
Begin
Insert into list phones(surname, fname, patronymic, tel_num)
Values (:fin_surname, :fin_fname, :fin_patronymic, :fin_tel_num);
End
Else
Begin
Update
list phones
set
surname=:fin_surname,
fname=:fin_fname,
patronymic=:fin_patronymic,
tel_num=:fin_tel_num
where
id=(:fin_id);
end
end
```

21. Скомпилировать процедуру.

22. Открыть SQLite Expert. Изучить команды меню и панель инструментов.

23. Создать базу данных «My BD».

24. Создать таблицу Students, в которой присутствуют следующие поля FIO, kurs, grupp.
25. Открыть IBExpert. Изучить команды меню и панель инструментов.
26. Зарегистрировать базу данных «Mu BD»
27. Создать таблицу Students, с полями FIO, Kurs, Grupp.

Контрольные вопросы:

1. Перечислите типы баз данных.
2. Что такое встраиваемые СУБД?
3. Как происходит доступ к данным?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 2

Тема: Построение семантических моделей

Цель: ознакомиться с основными объектами, составляющими инфологические модели; сформировать навыки по разработке инфологических моделей «сущность – связь».

Теоретический материал:

Метод сущность-связь называют также методом "ER-диаграмм": во-первых, ER –аббревиатура от слов Essence (сущность) и Relation (связь), во-вторых, метод основан на использовании диаграмм, называемых соответственно диаграммами ER-экземпляров и диаграммами ER-типа.

Основными понятиями метода сущность-связь являются следующие:

*сущность – представляет собой объект, информация о котором хранится в БД. Экземпляры сущности отличаются друг от друга и однозначно идентифицируются. Названиями сущностей являются, как правило, существительные, например: ПРЕПОДАВАТЕЛЬ, ДИСЦИПЛИНА, ГРУППА.

*Атрибут сущности – представляет собой свойство сущности. Это понятие аналогично понятию атрибута в отношении. Так, атрибутами сущности ПРЕПОДАВАТЕЛЬ может быть его Фамилия, Должность, Стаж (преподавательский) и т. д.

*Ключ сущности – атрибут или набор атрибутов, используемый для идентификации экземпляра сущности. Как видно из определения, понятие ключа сущности аналогично понятию ключа отношения.;

*Связь между сущностями. Связь двух или более сущностей - предполагает зависимость между атрибутами этих сущностей. Название связи обычно представляется глаголом. Примерами связей между сущностями являются следующие- ПРЕПОДАВАТЕЛЬ ВДЕТ ДИСЦИПЛИНУ (Иванов ВЕДЕТ "Организацию БД и знаний"), ПРЕПОДАВАТЕЛЬ ПРЕПОДАЕТ В ГРУППЕ (Иванов ПРЕПОДАЕТ В 6 группе);

*Степень связи – является характеристикой связи между сущностями, которая может быть следующих видов: 1:1, 1:M, M:1, M:M.;

*Класс принадлежности (КП) экземпляров сущности. КП сущности может быть: обязательным и необязательным. Класс принадлежности сущности является обязательным, если все экземпляры этой сущности обязательно участвуют в рассматриваемой связи, в противном случае класс принадлежности сущности является необязательным.

*Диаграммы ER-экземпляров;

*Диаграммы ER-типа.

Приведенные определения сущности и связи не полностью формализованы, но приемлемы для практики. Следует иметь в виду, что в результате проектирования могут быть получены несколько вариантов одной и той же БД. Так, два разных проектировщика, рассматривая одну и ту же проблему с разных точек зрения, могут получить различные наборы сущностей и связей.

При этом оба варианта могут быть рабочими, а выбор лучшего из них будет результатом личных предпочтений.

Диаграммы ER-экземпляров и ER-типа

С целью повышения наглядности и удобства проектирования для представления сущностей, экземпляров сущностей и связей между ними используются следующие графические средства:

- *диаграммы ER-экзмпляров,
- *диаграммы ER-типа, или ER-диаграммы.

На рисунке 1 приведена диаграмма ER-экземпляров для сущностей ПРЕПОДАВАТЕЛЬ и ДИСЦИПЛИНА со связью ВЕДЕТ.

ПРЕПОДАВАТЕЛЬ		ВЕДЕТ		ДИСЦИПЛИНА
Иванов	●		●	СУБД
Петров	●		●	С++
Сидоров	●		●	Паскаль
Егоров	●		●	Delphi
Козлов	●		●	ИС

Рис. 1 Диаграмма ER-экземпляров.

Диаграмма ER-экземпляров показывает, какую конкретно дисциплину (СУБД, С++ и т.д.) ведет каждый из преподавателей. На рис. 2 представлена диаграмма ER-типа, соответствующая рассмотренной диаграмме ER-экземпляров.



Рис. 2 Диаграмма ER-типа.

На начальном этапе проектирования БД выделяются атрибуты, составляющие ключи сущностей.

На основе анализа диаграмм ER-типа формируются отношения проектируемой БД. При этом учитывается степень связи сущностей и класс их принадлежности, которые, в свою очередь, определяются на основе анализа диаграмм ER-экземпляров соответствующих сущностей.

Степень связи является характеристикой связи между сущностями, которая может быть типа: 1:1, 1:M, M:1, M:M.

Класс принадлежности (КП) сущности может быть: обязательным и необязательным. КП сущности является обязательным, если все экземпляры этой сущности обязательно участвуют в рассматриваемой связи, в противном случае КП сущности является необязательным.

Варьируя КП сущностей для каждого из названных типов связи, можно получить несколько вариантов диаграмм ER-типа. Рассмотрим примеры некоторых из них.

Пример 1. Связи типа 1:1 и необязательный класс принадлежности

В приведенной на рис. 1 диаграмме степень связи между сущностями 1:1, а класс принадлежности обеих сущностей необязательный. Действительно, из рисунка видно следующее:

*каждый преподаватель ведет не более одной дисциплины, а каждая дисциплина ведется не более чем одним преподавателем (степень связи 1:1);

*некоторые преподаватели не ведут ни одной дисциплины и имеются дисциплины, которые не ведет ни один из преподавателей (класс принадлежности обеих сущностей необязательный).

Пример 2. Связи типа 1:1 и обязательный класс принадлежности

На рисунке 3 приведены диаграммы, у которых степень связи между сущностями 1:1, а класс принадлежности обеих сущностей обязательный.

а) Ег-экземпляров

ПРЕПОДАВАТЕЛЬ	ВЕДЕТ	ДИСЦИПЛИНА
Иванов		СУБД
Петров		С++
Сидоров		Паскаль
Егоров		Delphi
Козлов		ИС

Диаграмма Ег-экземпляров для связи 1:1 и обязательным КП обеих сущностей.

б) Ег-типов



Диаграмма Ег-типа для связи 1:1 и обязательным КП обеих сущностей.

Рис. 3 Диаграммы для связи 1:1 и обязательным КП обеих сущностей (О:О)

В этом случае каждый преподаватель ведет одну дисциплину и каждая дисциплина ведется одним преподавателем.

Возможны два промежуточных варианта с необязательным классом принадлежности одной из сущностей.

Диаграммы Ег-типа графически изображаются следующим образом:

*обязательное участие в связи экземпляров сущности отмечается блоком с точкой внутри, смежным с блоком этой сущности (рис. 3).

*необязательное участие экземпляров сущности в связи – дополнительный блок к блоку сущности не пристраивается, а точка размещается на линии связи (рис.2).

*символы на линии связи указывают на степень связи.

*под каждым блоком, соответствующим некоторой сущности, указывается ее ключ, выделяемый подчеркиванием. Многоточие за ключевыми атрибутами означает, что возможны другие атрибуты сущности, но ни один из

них не может быть частью ее ключа. Эти атрибуты выявляются после формирования отношений.

На практике степень связи и класс принадлежности сущностей при проектировании БД определяется спецификой предметной области. Рассмотрим примеры вариантов со степенью связи 1:M или M:1.

Пример 3. Связь типа 1:M – каждый преподаватель может вести несколько дисциплин, но каждая дисциплина ведется одним преподавателем,

Пример 4. Связи типа M:1 – каждый преподаватель может вести одну дисциплину, но каждую дисциплину могут вести несколько преподавателей.

Примеры с типом связи 1:M или M:1 могут иметь ряд вариантов, отличающихся классом принадлежности одной или обеих сущностей. Обозначим обязательный класс принадлежности символом "О", а необязательный - символом "Н", тогда варианты для связи типа 1:M условно можно представить как: О–О, О–Н, Н–О, Н–Н. Для связи типа M:1 также имеются 4 аналогичных варианта.

Пример 5. Связи типа 1:M вариант Н-О

Каждый преподаватель может вести несколько дисциплин ИЛИ ни одной, но каждая дисциплина ведется одним преподавателем (рис. 4).

а) ER-экземпляров

ПРЕПОДАВАТЕЛЬ	ВЕДЕТ	ДИСЦИПЛИНА
		СУБД
Иванов		ПЛ/1
Петров		Паскаль
Сидоров		Delphi
Егоров		ИС
Козлов		C++
		Java

Диаграмма ER-экземпляров для связи типа 1:M варианта Н-О

б) ER-типов



Диаграмма ER-типа для связи типа 1:M варианта Н-О

Рис. 4 Диаграммы для связи типа 1:M варианта Н-О

Пример 6. Связи типа M: M

Каждый преподаватель может вести несколько дисциплин, а каждая дисциплина может вестись несколькими преподавателями. Как и в случае других типов связей, для связи типа M:M возможны 4 варианта, отличающиеся КП сущностей.

Пример 7. Связи типа M:M и вариант класса принадлежности О-Н

Допустим, что каждый преподаватель ведет не менее одной дисциплины, а дисциплина может вестись более чем одним преподавателем, есть и та-

кие дисциплины, которые никто не ведет. Соответствующие этому случаю диаграммы приведены на рисунке 5.

а) ER-экземпляров

ПРЕПОДАВАТЕЛЬ	ВЕДЕТ	ДИСЦИПЛИНА
		Средства СУБД
Иванов		ПЛ/1
Петров		Паскаль
Сидоров		Delphi
Егоров		ИС
Козлов		C++
		Java

Диаграмма ER-экземпляров для связи типа М:М и вариант класса принадлежности О-Н.

б) ER-типов



Диаграмма ER-типов для связи типа М : М и варианта О-Н.

Рис. 5 Диаграммы для связи типа М:М и варианта О-Н

Выявление сущностей и связей между ними, а также формирование на их основе диаграмм ER-типа выполняется на начальных этапах метода сущность-связь. Рассмотрим этапы реализации метода.

Задания:

1. Изучить основные понятия семантической модели.
2. Проанализировать примеры, представленные выше.
3. Выбрать из списка предметную область (приложение 1).
4. Выделить семантические сущности.

Контрольные вопросы:

1. Дайте определение следующим понятиям: сущность, атрибут, связь, ключ.
2. Какие типы связей Вы знаете?

Задание на дом:

Выполнить отчет по практической работе.

Предметные области (БД)

- | | |
|---------------------------|--|
| 1. Кадры | 2. Социальная работа |
| 3. Документы | 4. Аптека |
| 5. Библиотека | 6. Пенсионный фонд |
| 7. Лесничество | 8. Оптовый склад |
| 9. География | 10. Факультет |
| 11. Банки | 12. Кафедра |
| 13. Поликлиника | 14. Учебные планы |
| 15. Юриспруденция | 16. Картинная галерея |
| 17. Снабжение | 18. Музей |
| 19. Сбыт | 20. Продажа и регистрация авиабилетов |
| 21. Производство – детали | 22. Фонотека |
| 23. ГИБДД | 24. Видеотека |
| 25. Налоги | 26. Футбольные клубы |
| 27. Земельный кадастр | 28. Кинозвезды |
| 29. Расписание | 30. Баскетбольные клубы |
| 31. Учебный план | 32. Хоккеисты |
| 33. Учебный процесс | 34. Теннис |
| 35. Магазин | 36. Обеспеченность учебного процесса литературой |

Практическая работа № 3

Тема: Построение инфологических моделей

Цель: изучить основные объекты, составляющие инфологические модули; сформировать навыки по разработке инфологических моделей «сущность-связь».

Теоретический материал:

Пример разработки инфологической модели

В базе данных должна быть отражена информация из экзаменационной ведомости.

СТУДЕНТ СДАЕТ ЭКЗАМЕН ПО ДИСЦИПЛИНЕ ПРЕПОДАВАТЕЛЮ.

1. Выделим сущности: СТУДЕНТ, ДИСЦИПЛИНА, ПРЕПОДАВАТЕЛЬ.

2. Выделим связи между сущностями.

Модель с тернарной связью.

Связь – СДАЕТ ЭКЗАМЕН – связывает три сущности.

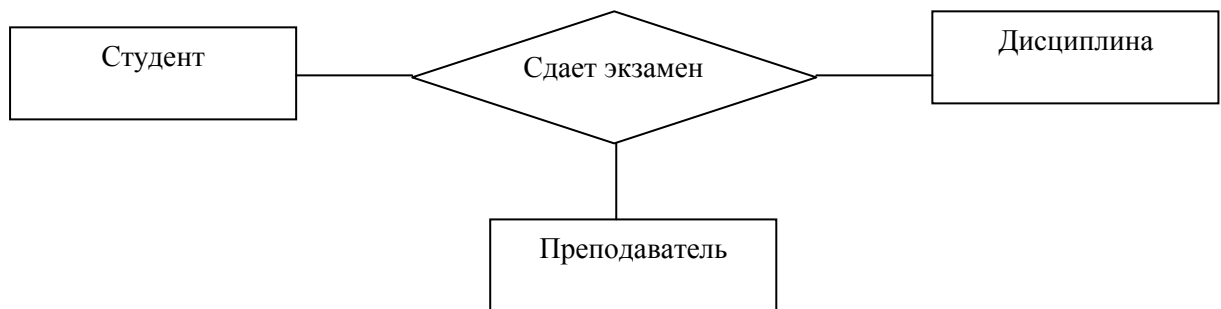


Рис. 1 Модель с тернарной связью

Модель с бинарными связями.

Связи:

СДАЕТ ЭКЗАМЕН – связывает сущности СТУДЕНТ и ДИСЦИПЛИНА;

ПРИНИМАЕТ ЭКЗАМЕН – связывает сущности ПРЕПОДАВАТЕЛЬ и ДИСЦИПЛИНА.

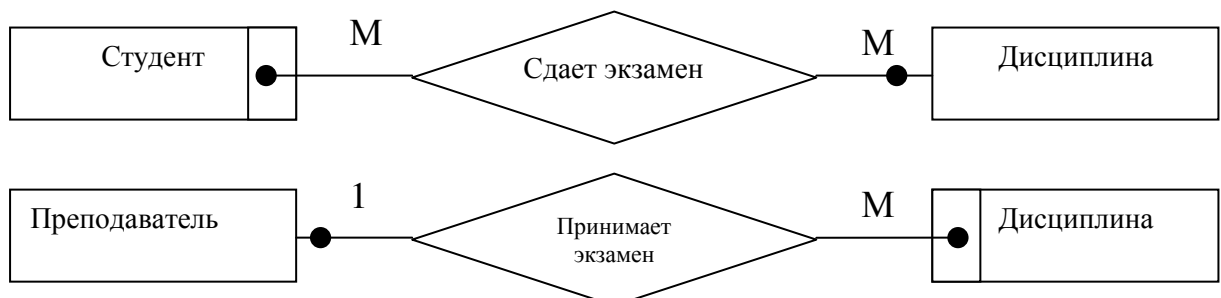


Рис. 2 Модели бинарных связей

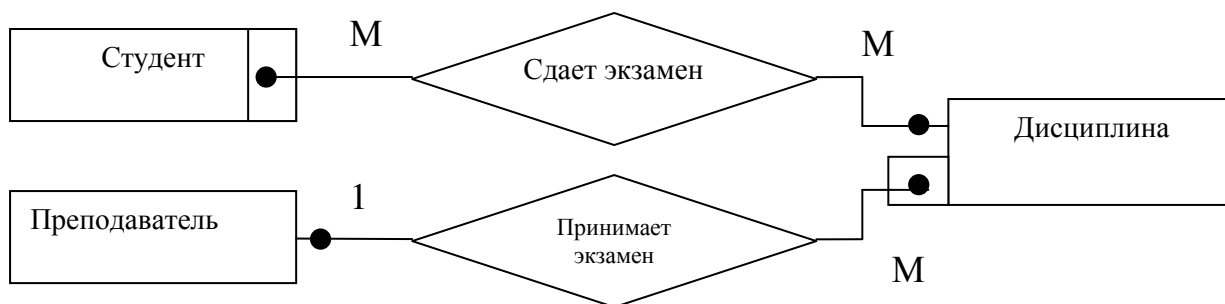


Рис. 3 Объединенная модель с бинарными связями

3. Выделим атрибуты сущностей и связей.

Атрибуты сущности СТУДЕНТ: Фамилия, Имя, Отчество, Группа, Номер зачетки.

Атрибуты сущности ДИСЦИПЛИНА: Название, Семестр, Число часов.

Атрибуты сущности ПРЕПОДАВАТЕЛЬ: Фамилия, Имя, Отчество, Степень, Звание.

Модель с тернарной связью.

Атрибуты связи СДАЕТ ЭКЗАМЕН: Оценка, Дата, Номер экзаменационной ведомости.

Модель с бинарными связями.

Атрибуты связи СДАЕТ ЭКЗАМЕН: Оценка, Дата, Номер экзаменационной ведомости.

Связь ПРИНИМАЕТ ЭКЗАМЕН не имеет атрибутов.

Задание:

1. Для каждой выделенной семантической сущности (практическая работа № 2) определить набор атрибутов.
2. Определить связи между семантическими сущностями.
3. Построить Er-диаграммы для выбранной предметной области.
4. Начертить ER-диаграммы в редакторе векторной графики (например, OpenOffice.Draw) или в текстовом процессоре (например, Microsoft Word).

Контрольные вопросы:

1. Назовите типы связей?
2. Что такое атрибут сущности?
3. Что содержит ER-диаграмма?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 4

Тема: Реляционная модель данных

Цель: формирование общих представлений о создании реляционной модели данных.

Теоретический материал:

Реляционная модель данных – логическая модель данных. В настоящее время эта модель является фактическим стандартом, на который ориентируются практически все современные коммерческие СУБД.

Структура реляционной модели данных:

- структурная
- манипуляционная
- целостная

Структурная часть модели определяет, то что единственной структурой данных является нормализованное n -арное отношение. Отношения удобно представлять в форме таблиц, где каждая строка есть кортеж, а каждый столбец – атрибут, определенный на некотором домене. Реляционная база данных представляет собой конечный набор таблиц.

Манипуляционная часть модели определяет два фундаментальных механизма манипулирования данными – реляционная алгебра и реляционное исчисление. Основной функцией манипуляционной части реляционной модели является обеспечение меры реляционности любого конкретного языка реляционных БД.

Язык называется реляционным, если он обладает не меньшей выразительностью и мощностью, чем реляционная алгебра или реляционное исчисление.

Целостная часть модели определяет требования *целостности сущностей* и *целостности ссылок*. Первое требование состоит в том, что любое отношение должно обладать первичным ключом. Требование целостности по ссылкам, или требование внешнего ключа состоит в том, что для каждого значения внешнего ключа, появляющегося в ссылающемся отношении, в отношении, на которое ведет ссылка, должен найтись кортеж с таким же значением первичного ключа, либо значение внешнего ключа должно быть неопределенным (т.е. ни на что не указывать).

Структура реляционной модели данных

Можно провести аналогию между элементами реляционной модели данных и элементами модели "сущность-связь". Реляционные отношения соответствуют наборам сущностей, а кортежи – сущностям. Поэтому, также как и в модели "сущность-связь" столбцы в таблице, представляющей реляционное отношение, называют атрибутами.

Задание:

1. Создать новую базу данных.
2. Создать таблицу базы данных.
3. Определить поля таблицы в соответствии с таблицей 1.
4. Сохранить созданную таблицу.

Таблица 1 «Преподаватели»

Имя поля	Тип данных	Размер поля
Код преподавателя	Счетчик	
Фамилия	Текстовый	15
Имя	Текстовый	15
Отчество	Текстовый	15
Дата рождения	Дата/время	Краткий
Должность	Текстовый	9
Дисциплина	Текстовый	11
Телефон	Текстовый	9
Зарплата	Денежный	

5. Ввести ограничения на данные, вводимые в поле «Должность»; должны вводиться только слова Профессор, Доцент или Ассистент.
6. Задать текст сообщения об ошибке, который будет появляться на экране при вводе неправильных данных в поле «Должность».
7. Задать значение по умолчанию для поля «Должность» в виде слова Доцент.
8. Ввести ограничения на данные в поле «Код преподавателя»; эти данные не должны повторяться.
9. Создать столбец подстановок для поля «Дисциплина».
10. Заполнить таблицу произвольными данными (не менее 10 записей) и проверить реакцию системы на ввод неправильных данных в поле «Должность».
11. Изменить ширину каждого поля таблицы в соответствии с шириной данных.
12. Произвести поиск в таблице преподавателя Миронова.
13. Произвести замену данных: к примеру, изменить заработную плату ассистенту Сергеевой с 4500 р. на 4700 р.
14. Произвести сортировку данных в поле «Год рождения» по убыванию.
15. Произвести фильтрацию данных по полям «Должность» и «Дисциплина».
16. Просмотреть созданную таблицу, как она будет выглядеть на листе бумаги при печати.
17. Создать таблицу «Студенты»

Имя поля	Тип данных	Размер поля
Код студента	Счетчик	
Фамилия	Текстовый	25
Имя	Текстовый	25

Отчество	Текстовый	25
Дата рождения	Дата/время	

18. Заполнить таблицу (не менее 15 записей).

19. Создать таблицу «Сессия»

Имя поля	Тип данных	Размер поля
Код преподавателя	Числовой	
Код студента	Числовой	
Дисциплина	текстовый	20
Оценка	Числовой	

20. Заполнить таблицу данными (не менее 15).

21. Установить связи между таблицами.

22. С помощью Мастера форм создать форму Состав преподавателей.

23. Найти запись о доценте Гришине, находясь в режиме формы.

24. Изменить зарплату ассистенту Сергеевой с 4500 р. на 4900 р.

25. Произвести сортировку данных в поле «Фамилия» по убыванию.

26. Произвести фильтрацию данных по полю «Должность».

27. Изменить название поля «Дисциплина» на «Преподаваемая дисциплина».

28. На основе таблицы Преподаватели создать простой запрос на выборку, в котором должны отображаться фамилии, имена, отчества преподавателей и их должность.

29. Данные запроса отсортировать по должностям.

30. Сохраните запрос.

31. Создать запрос на выборку с параметром, в котором должны отображаться фамилии, имена, отчества преподавателей и преподаваемые ими дисциплины, а в качестве параметра задать фамилию преподавателя и выполните этот запрос для преподавателя Гришина.

32. На основе таблицы Преподаватели создать отчет с группированием данных по должностям.

Контрольные вопросы:

1. Что такое реляционная модель данных?
2. Какова структура реляционной модели данных?
3. Опишите процесс создания базы данных в Access?
4. Какие типы данных задаются в Access?
5. В чем разница между сортировкой и фильтрацией данных?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 5

Тема: Преобразование инфологической модели данных в реляционную

Цель: сформировать навыки по преобразованию инфологической модели предметной области в реляционную.

Теоретический материал:

Переход от инфологической модели "сущность-связь"- это сравнительно простая задача, поскольку в терминологии и принципах ER-модели и реляционного подхода имеется взаимно однозначное соответствие. Существует ряд хорошо зарекомендовавших себя правил с помощью которых из ER-диаграмм откроются реляционные таблицы:

1. Каждая простая сущность превращается в таблицу. Простая сущность - сущность, не являющаяся подтипом и не имеющая подтипов. Имя сущности становится именем таблицы.

2. Каждый атрибут становится возможным столбцом с тем же именем; может выбираться более точный формат. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

3. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если имеется несколько возможных уникальных идентификатора, выбирается наиболее используемый. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи (этот процесс может продолжаться рекурсивно). Для именования этих столбцов используются имена концов связей и/или имена сущностей.

4. Связи многие-к-одному (и один-к-одному) становятся внешними ключами. Т.е. делается копия уникального идентификатора с конца связи "один", и соответствующие столбцы составляют внешний ключ. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

5. Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается в основном базировать запросы.

6. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:

- все подтипы в одной таблице (а)
- для каждого подтипа - отдельная таблица (б).

При применении способа (а) таблица создается для наиболее внешнего супертипа, а для подтипов могут создаваться представления. В таблицу добавляется, по крайней мере, один столбец, содержащий код ТИПА; он становится частью первичного ключа.

При использовании метода (б) для каждого подтипа первого уровня (для более нижних - представления) супертип воссоздается с помощью представления UNION (из всех таблиц подтипов выбираются общие столбцы - столбцы супертипа).

Все в одной таблице	Таблица – на подтип
<i>Преимущества</i>	
Все хранится вместе	Более ясны правила подтипов
Легкий доступ к супертипу и подтипам	Программы работают только с нужными таблицами
Требуется меньше таблиц	
<i>Недостатки</i>	
Слишком общее решение	Слишком много таблиц
Требуется дополнительная логика работы с разными наборами столбцов и разными ограничениями	Смущающие столбцы в представлении UNION
Потенциальное узкое место (в связи с блокировками)	Потенциальная потеря производительности при работе через UNION
Столбцы подстановок должны быть необязательными	Над супертипом невозможны модификации
В некоторых СУБД для хранения неопределенных значений требуется дополнительная память	

7. Имеется два способа работы при наличии исключаящих связей:

- общий столбец (а)
- явные внешние ключи (б).

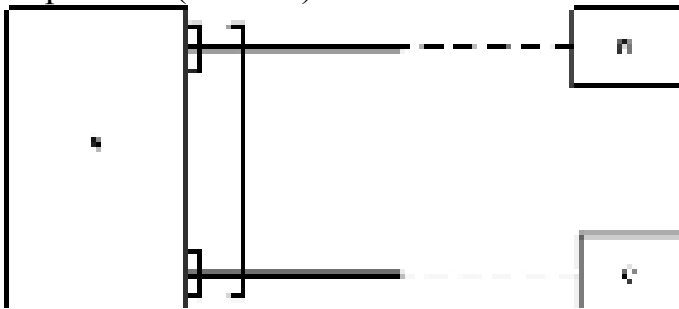
Если остающиеся внешние ключи все в одном домене, т.е. имеют общий формат (способ (а)), то создаются два столбца: идентификатор связи и идентификатор сущности. Столбец идентификатора связи используется для различения связей, покрываемых дугой исключения. Столбец идентификатора сущности используется для хранения значений уникального идентификатора сущности на дальнем конце соответствующей связи.

Если результирующие внешние ключи не относятся к одному домену, то для каждой связи, покрываемой дугой исключения, создаются явные столбцы внешних ключей; все эти столбцы могут содержать неопределенные значения.

Общий домен	Явные внешние ключи
<i>Преимущества</i>	
Нужно только два столбца	Условия соединения - явные
<i>Недостатки</i>	
Оба дополнительных атрибута должны использоваться в соединениях	Слишком много столбцов

Альтернативные модели сущностей:

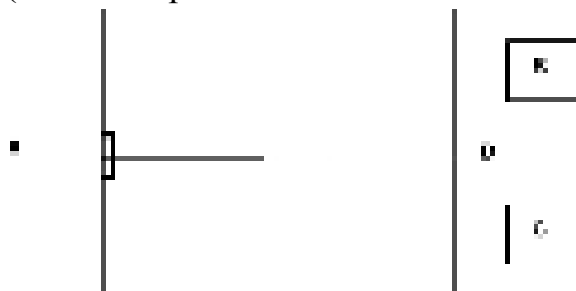
Вариант 1 (плохой)



Вариант 2 (существенно лучше, если подтипы действительно существуют)



Вариант 3 (годится при наличии осмысленного супертипа D).



Задания:

1. Преобразовать полученную в практической работе 5 инфологическую модель предметной области в реляционную.

Контрольные вопросы:

1. Перечислите базовые свойства реляционной модели данных?
2. Укажите базовые свойства инфологической модели данных?
3. Какие требования предъявляются к структурной части реляционной модели данных?
4. Какие требования предъявляются к манипуляционной части реляционной модели данных?
5. Какие требования предъявляются к целостной части реляционной модели данных?
6. Перечислите основные этапы преобразования инфологической модели данных в реляционную.

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 6

Тема: Язык запросов SQL

Цель: освоить технологию создания запросов с помощью структурированного языка запросов SQL.

Теоретический материал:

Основы языка SQL

SQL (Structured Query Language) – это структурированный язык запросов к реляционным базам данных (БД). SQL является декларативным языком, основанным на операциях реляционной алгебры. Существуют два стандарта SQL, определённые американским национальным институтом стандартов (ANSI): SQL-89 (SQL-1) и SQL-92 (SQL-2). В настоящее время разрабатывается новый стандарт – SQL-3. Большинство коммерческих систем управления базами данных (СУБД) поддерживают стандарт SQL-92, который принят ISO (International Standards Organization) в качестве международного стандарта. Многие версии имеют свои отличия, которые касаются, в основном, синтаксиса.

Язык работы с базами данных должен предоставлять пользователям следующие возможности:

- создавать базу данных и таблицы с полным описанием их структуры;
- выполнять основные операции манипулирования данными (добавление, изменение, удаление данных);
- выполнять запросы, осуществляющие преобразование данных в необходимую информацию.

Для реализации этих функций SQL включает 4 группы средств:

- DDL (Data Definition Language) – язык определения данных;
- DML (Data Manipulation Language) – язык манипулирования данными;
- DQL (Data Query Language) – язык запросов данных;
- DCL (Data Control Language) – язык управления данными.

По стандарту ANSI DCL является частью DDL.

В командах SQL не различаются прописные и строчные буквы (за исключением строк). Каждая команда заканчивается символом ';'. Значения параметров по умолчанию выделено подчеркиванием, например, ALL.

Примем следующие обозначения для описания синтаксиса:

{ } – содержимое скобок рассматривается как единое целое для остальных символов;

| – заменяет слово ИЛИ;

[] – содержимое этих скобок является необязательным;

... – всё, что предшествует этим символам, может повторяться произвольное число раз;

... – всё, что предшествует этим символам, может повторяться произвольное число раз, каждое вхождение отделяется запятой.

Существуют три формы SQL: интерактивный (*Interactive*), статический (*Static*) и динамический (*Dynamic*). Функционируют они одинаково, но используются по-разному.

- * Интерактивный SQL применяется для непосредственной работы с БД - пользователь вводит SQL-оператор, он сразу же выполняется и пользователь видит результат выполнения (или код ошибки).

- * Статический SQL содержит SQL-операторы, жестко закодированные в теле исполняемого приложения. Наиболее распространен встроенный SQL (*Embedded SQL*), где SQL-код включается в исходный текст (базовой) программы, написанной на другом языке (например, C или Pascal); при использовании встроенного SQL результаты выполнения операторов SQL перенаправляются в переменные, которыми оперирует базовая программа. К настоящему времени SQL встроен в языки Ada, Cobol, Fortran, C, Pascal, PL/1, Java, Mumps (теперь M).

- * Динамический SQL также является частью приложения, но конкретный SQL-код генерируется во время выполнения (Run Time), а не вводится заранее.

Фактически везде описывается интерактивная форма SQL - сначала приводится текст SQL-запроса, а ниже дается ответ исполняющей системы (обычно в виде таблицы).

Типы данных SQL

- Символьные типы данных - содержат буквы, цифры и специальные символы.

- **CHAR** или **CHAR(n)** -символьные строки фиксированной длины. Длина строки определяется параметром **n**. **CHAR** без параметра соответствует **CHAR(1)**. Для хранения таких данных всегда отводится **n** байт вне зависимости от реальной длины строки.

- **VARCHAR(n)** - символьная строка переменной длины. Для хранения данных этого типа отводится число байт, соответствующее реальной длине строки.

- Целые типы данных - поддерживают только целые числа (дробные части и десятичные точки не допускаются). Над этими типами разрешается выполнять арифметические операции и применять к ним агрегирующие функции (определение максимального, минимального, среднего и суммарного значения столбца реляционной таблицы).

- **INTEGER** или **INT**- целое, для хранения которого отводится, как правило, 4 байта. (Замечание: число байт, отводимое для хранения того или иного числового типа данных зависит от используемой СУБД и аппаратной платформы, здесь приводятся наиболее "типичные" значения) Интервал значений от - 2147483647 до + 2147483648

- **SMALLINT** - короткое целое (2 байта), интервал значений от - 32767 до +32768

- Вещественные типы данных - описывают числа с дробной частью.

- **FLOAT** и **SMALLFLOAT** - числа с плавающей точкой (для хранения отводится обычно 8 и 4 байта соответственно).

- **DECIMAL(p)** - тип данных аналогичный **FLOAT** с числом значащих цифр **p**.

- **DECIMAL(p,n)** - аналогично предыдущему, **p** - общее количество десятичных цифр, **n** - количество цифр после десятичной запятой.

- Денежные типы данных - описывают, естественно, денежные величины. Если в ваша система такого типа данных не поддерживает, то используйте **DECIMAL(p,n)**.

- **MONEY(p,n)** - все аналогично типу **DECIMAL(p,n)**. Вводится только потому, что некоторые СУБД предусматривают для него специальные методы форматирования.

- Дата и время - используются для хранения даты, времени и их комбинаций. Большинство СУБД умеет определять интервал между двумя датами, а также уменьшать или увеличивать дату на определенное количество времени.

- **DATE** - тип данных для хранения даты.

- **TIME** - тип данных для хранения времени.

- **INTERVAL** - тип данных для хранения верменного интервала.

- **DATETIME** - тип данных для хранения моментов времени (год + месяц + день + часы + минуты + секунды + доли секунд).

- Двоичные типы данных - позволяют хранить данные любого объема в двоичном коде (оцифрованные изображения, исполняемые файлы и т.д.). Определения этих типов наиболее сильно различаются от системы к системе, часто используются ключевые слова:

- **BINARY**

- **BYTE**

- **BLOB**

- Последовательные типы данных - используются для представления возрастающих числовых последовательностей.

- **SERIAL** - тип данных на основе **INTEGER**, позволяющий сформировать уникальное значение (например, для первичного ключа). При добавлении записи СУБД автоматически присваивает полю данного типа значение, получаемое из возрастающей последовательности целых чисел.

В заключение следует сказать, что для всех типов данных имеется общее значение **NULL** - "не определено". Это значение имеет каждый элемент столбца до тех пор, пока в него не будут введены данные. При создании таблицы можно явно указать СУБД могут ли элементы того или иного столбца иметь значения **NULL** (это не допустимо, например, для столбца, являющегося первичным ключом).

Задания:

Ситуация 1. Исходными являются три отношения R1, R2 и R3. Все они имеют эквивалентные схемы.

- $R_1 = (\text{ФИО}, \text{Паспорт}, \text{Школа})$;
- $R_2 = (\text{ФИО}, \text{Паспорт}, \text{Школа})$;
- $R_3 = (\text{ФИО}, \text{Паспорт}, \text{Школа})$.

Рассмотрим ситуацию поступления в высшие учебные заведения, которая была характерна для периода, когда были разрешены так называемые репетиционные вступительные экзамены, которые сдавались раньше основных вступительных экзаменов в вуз.

Отношение R_1 содержит список абитуриентов, сдававших репетиционные экзамены.

Отношение R_2 содержит список абитуриентов, сдававших экзамены на общих условиях.

Отношение R_3 содержит список абитуриентов, принятых в институт.

Будем считать, что при неудачной сдаче репетиционных экзаменов абитуриент мог делать вторую попытку и сдавать экзамены в общем потоке, поэтому некоторые абитуриенты могут присутствовать как в первом, так и во втором отношении.

Задача 1. Определить список абитуриентов, которые поступали два раза и не поступили в вуз.

Задача 2. Определить список абитуриентов, которые поступили в вуз с первого раза, то есть они сдавали экзамены только один раз и сдали их так хорошо, что сразу были зачислены в вуз.

Задача 3. Определить список абитуриентов, которые поступили в вуз только со второго раза.

Задача 4. Определить список абитуриентов, которые поступали только один раз и не поступили.

Ситуация 2. Даны три отношения R_1 , R_2 и R_3 .

- $R_1 = (\text{ФИО}, \text{Номер_зач})$;
- $R_2 = (\text{Дисциплина})$;
- $R_3 = (\text{Номер_зач}, \text{Дисциплина})$.

Рассмотрим ситуацию сдачи экзамена студентами одной специальности.

Отношение R_1 содержит список студентов, которые должны сдавать экзамены.

Отношение R_2 содержит список всех дисциплин, по которым студенты должны сдавать экзамены.

Отношение R_3 содержит список студентов, сдавших экзамены.

Задача 5. Сформировать список фамилий студентов, которые сдавали экзамены.

Задача 6. Сформировать список фамилий студентов, которые не сдали хотя бы один экзамен.

Задача 7. Определить список ФИО абитуриентов, сдавших все экзамены.

Задача 8. Определить список ФИО абитуриентов, сдавших все экзамены на 4 и 5.

Ситуация 3. Рассмотрим набор отношений, которые моделируют сдачу сессии студентами некоторого учебного заведения.

■ $R_1 = \langle \text{ФИО}, \text{Дисциплина}, \text{Оценка} \rangle;$

■ $R_2 = \langle \text{ФИО}, \text{Группа} \rangle;$

■ $R_3 = \langle \text{Группы}, \text{Дисциплина} \rangle,$

где R_1 — информация о попытках (как успешных, так и неуспешных) сдачи экзаменов студентами;

R_2 — состав групп;

R_3 — список дисциплин, которые надо сдавать каждой группе.

Задача 9. Определить список фамилий студентов, которые сдали экзамен по «БД» на «отлично».

Задача 10. Определить список тех, кто должен был сдавать экзамен по БД, но пока еще не сдавал.

Задача 11. Определить список несчастных, имеющих несколько двоек:

Задача 12. Определить список круглых отличников.

Контрольные вопросы:

1. Укажите, какое назначение языка SQL.

2. Что такое перекрестный запрос?

3. Что такое запрос на объединение?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 7

Тема: Язык запросов SQL

Цель: закрепить навыки по созданию запросов с помощью SQL.

Задания:

Задание 1

Даны отношения, моделирующие работу банка и его филиалов. Клиент может иметь несколько счетов, при этом они могут быть размещены как в одном, так и в разных филиалах банка. В отношении R_1 содержится информация обо всех клиентах и их счетах в филиалах нашего банка. Каждый клиент, в соответствии со своим счетом, может рассчитывать на некоторый кредит от нашего банка, сумма допустимого кредита также зафиксирована.

- $R_1=(\text{ФИО клиента, № филиала, № счета, Остаток, Кредит});$
- $R_2=(\text{№ филиала, Район})$

С использованием языка реляционной алгебры составить запросы, позволяющие выбрать:

1. Филиалы, клиенты которых имеют счета с остатком, превышающим \$1000.
2. Клиентов, которые имеют счета во всех филиалах данного банка.
3. Клиентов, которые имеют только по одному счету в разных филиалах банка. То есть в общем у этих клиентов может быть несколько счетов, но в одном филиале не более одного счета.
4. Клиенты, которые имеют счета в нескольких филиалах банка, расположенных только в одном районе.
5. Филиалы, которые не имеют ни одного клиента.
6. Филиалы, которые имеют клиентов с остатком на счету 0 (ноль).
7. Филиалы, у которых есть клиенты с кредитом, превышающим остаток на счету в 2 раза.

Задание 2

Даны отношения, моделирующие работу международной фирмы, имеющей несколько филиалов. Филиалы фирмы могут быть расположены в разных странах, это отражено в отношении R_1 . Клиенты фирмы также могут быть из разных стран, и это отражено в отношении R_4 . По каждому конкретному заказу клиент мог заказать несколько разных товаров.

- $R_1=(\text{Филиал, Страна});$
- $R_2=(\text{Филиал, Заказчик, № заказа});$
- $R_3=(\text{№ заказа, Товар, Количество});$
- $R_4=(\text{Заказчик, Страна});$

С использованием реляционной алгебры составить запросы, позволяющие выбрать:

1. Заказчиков, которые работают со всеми филиалами фирмы, но покупают только один товар.
2. Филиалы фирмы, которые торгуют всеми товарами.
3. Товары, которые фирма продает только в одной стране.

4. Заказчиков, которые работают с филиалами фирмы, которые расположены только в одной стране.
5. Филиалы, с которыми не работает ни один заказчик.
6. Заказчиков, которые работают только с филиалами, расположенными в той же стране, что и заказчик.
7. Заказчиков, которые покупают все товары, представленные в отношении R_3 .

Задание 3

Даны отношения, моделирующие работу фирмы, занимающейся разработкой программных систем. Каждый сотрудник административно закреплен только за одним отделом. Файлы хранятся на разных серверах. На разных серверах файлы могут иметь одинаковые имена. Создатель файла является его владельцем, поэтому у каждого файла только один владелец, но владелец файла может разрешить пользоваться файлом другим сотрудникам. Существует множество системного программного обеспечения, каждая программа может работать с одним или с несколькими файлами, расположенными на одном или нескольких серверах:

- R_1 =(Название файла, Имя владельца файла);
- R_2 =(Название программы, название файла, Сервер);
- R_3 =(Название файла, Название сервера);
- R_4 =(Сотрудник, Отдел);

С использованием реляционной алгебры и языка составить запросы, позволяющие выбрать:

1. Файлы, которые имеют нескольких пользователей из разных отделов.
2. Программы, которые работают только с одним файлом.
3. Файлы, которые имеют одно и тоже имя, но расположены на различных серверах и используются сотрудниками разных отделов.
4. Файлы, с которыми работают сотрудники всех отделов.
5. Файлы, пользователями которых являются сотрудники только одного отдела.
6. Программы, которые работают со всеми серверами.
7. Отделы, сотрудники которых не работают ни с одним файлом. То есть отделы, в которых нет ни одного сотрудника, работающего с каким-нибудь файлом.
8. Отделы, сотрудники которых работают со всеми серверами.
9. Серверы, с которыми работают сотрудники только одного отдела.

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 8

Тема: Проектирование баз данных

Цель работы: освоение общих принципов проектирования реляционных моделей данных, знакомство с основами реляционного исчисления.

Теоретический материал:

Для проектирования баз данных используют следующие инструменты:

1. Реляционная модель данных – способ представления данных предметной области
2. Структурированный язык запросов (SQL) – универсальный способ манипулирования данными.

При разработке базы данных выделяют следующие уровни моделирования, при помощи которых происходит переход от предметной области к конкретной реализации базы данных (БД) средствами конкретной системы управления базами данных (СУБД):

1. Модель предметной области
2. Логическая модель данных
3. Физическая модель данных
4. База данных и приложения

При разработке логической модели данных выделяют два подхода:

1. Сбор информации об объектах решаемой задачи в рамках одной таблицы (одного отношения) и последующая декомпозиция ее на несколько взаимосвязанных таблиц на основе процедуры нормализации отношений.
2. Формулирование знаний о системе (определение типов исходных данных и их взаимосвязей) и требований к обработке данных, получение с помощью CASE-системы (Computer Aided Software Engineering –система автоматизации проектирования и разработки баз данных) готовой схемы БД или даже готовой прикладной информационной системы.

Задания:

1. Самостоятельно разработать схему данных для предметной области (формирование описания таблиц, определение первичных ключей).

Требования:

- схема данных должна содержать не менее 5 базовых отношений (не считая справочных);
 - 2 запроса в терминах реляционного исчисления;
2. По выбранной предметной области спроектировать ER-диаграмму:
 - провести преобразование ER-модели в реляционную модель.
 - количество сущностей в ER-диаграмме не менее 7;

- между сущностями в ER-диаграмме должны быть показаны все типы связей.

Варианты задания

1. «Абитуриент». а) администратор ВУЗа; б) член технической приёмной комиссии; в) член экзаменационной комиссии.
2. «Факультет». а) ректор; б) декан; в) преподаватель; г) студент.
3. «Супермаркет». а) заведующий; б) продавец; в) покупатель; г) снабженец.
4. «Ателье по ремонту бытовой техники». а) директор; б) мастер; в) клиент; г) поставщик деталей.
5. «Домоуправление». а) руководитель; б) паспортист; в) бригадир ремонтников; г) работник районной администрации.
6. «Общественный транспорт». а) руководитель предприятия; б) диспетчер; в) водитель; г) пассажир.
7. «Библиотека». а) библиотекарь; б) читатель; в) работник архива.
8. «Общественное питание». а) руководитель; б) снабженец; в) повар; г) посетитель.
9. «Служба занятости». а) регистратор безработных; б) администратор общественных работ; в) администратор по переобучению; г) безработный.
10. «Овощная база». а) руководитель базы; б) поставщик; в) заведующий магазином; г) диспетчер автотранспорта.
11. «Обслуживание пассажиров на ж/д вокзале». а) администратор; б) кассир; в) служба грузодоставки; г) пассажир.
12. «Дом отдыха». а) администратор дома отдыха; б) представитель профкома предприятия; в) клиент; г) заведующий столовой при доме отдыха.
13. «Грузоперевозки». а) отправитель; б) получатель; в) диспетчер; г) водитель автотранспорта.
14. «Школа». а) директор; б) учитель; в) родитель; г) ученик.
15. «Чемпионат по футболу». а) директор стадиона; б) судья; в) администратор команды; г) болельщик.
16. «Туристическая фирма». а) руководитель фирмы; б) менеджер; в) клиент.
17. «Фотоателье». а) руководитель; б) клиент; в) фотограф.
18. «Музей». а) экскурсовод; б) билетер; в) работник хранилища; г) составитель экспозиций; д) посетитель музея.
19. «Рекламное агентство». а) руководитель агентства; б) рекламода-тель; в) менеджер агентства.
20. «Поликлиника». а) врач; б) больной; в) работник регистратуры.
21. «Студия звукозаписи». а) звукорежиссер; б) исполнитель; в) бухгалтер.
22. «Коллекционный винный погреб». а) владелец; б) сомми-лье(хранитель винного погреба); в) покупатель.
23. «Оператор сотовой связи». а) абонент; б) менеджер; в) работник технического отдела; г) работник справочной службы.

24. «Паспортный стол». а) начальник ПС; б) паспортист; в) гражданин; г) работник справочной службы.

25. «Автосалон». а) заказчик; б) поставщик; в) менеджер; г) работник технического сервиса.

26. «Выставочная галерея». а) посетитель выставки; б) экскурсовод; в) куратор галереи г) составитель выставки.

27. «Политические деятели страны». а) избиратель; б) работник центральной избирательной комиссии.

28. «Торговля недвижимостью» а) риэлтер; б) продавец; в) покупатель.

29. «Аптека» а) фармацевт; б) поставщик; в) покупатель.

Контрольные вопросы:

- 1) Охарактеризуйте основные понятия базы данных.
- 2) Раскройте понятие модели данных.
- 3) Перечислите виды моделей данных, проведите их сравнительный анализ.
- 4) Дайте характеристику основным элементам структуры данных.
- 5) Что такое атрибут сущности?
- 6) Как определяется первичный ключ таблицы?
- 7) Что ставится в соответствие многозначным атрибутам?
- 8) Как выполняется преобразование составного атрибута сущности?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 9

Тема: Создание таблиц, ввод и редактирование данных в MS Office Access 2007

Цель работы: освоить технологию создания базы данных, таблиц; научиться осуществлять ввод и редактирование данных в СУБД MS Office Access 2007.

Теоретический материал:

Microsoft Office Access 2007 данные организуются в таблицы. Большинство БД включают несколько таблиц. Например, в одной таблице могут храниться сведения о продуктах, во второй — сведения о заказах, а в третьей — сведения о клиентах.

Каждая строка называется также записью, а каждый столбец, или тип элемента, называется также полем.

1. Компоненты БД MS Access

Таблица содержит данные по определенной теме, например, сведения о сотрудниках или товарах. Каждая запись в таблице включает данные об одном элементе, например о конкретном сотруднике. Запись состоит из полей и включает такие сведения, как имя, адрес и телефонный номер. Кроме того, запись обычно называется строкой, а поле – столбцом.

База данных может включать множество таблиц, в которых хранятся данные по различным темам. Каждая таблица может состоять из множества полей различного типа, включая текст, числа, даты и рисунки.

Формы иногда называются окнами ввода данных. Это интерфейсы, которые используются для работы с данными и часто содержат кнопки для выполнения различных команд. Большинство пользователей баз данных предпочитают просматривать, вводить и редактировать данные таблиц при помощи форм.

Формы позволяют работать с данными в удобном формате; кроме того, в них можно добавлять функциональные элементы, например кнопки команд.

Отчеты служат для сбора и представления данных, содержащихся в таблицах. Каждый отчет можно отформатировать так, чтобы представить сведения в наиболее удобном виде.

Обычно отчеты форматируют для печати, но их можно также просматривать на экране, экспортировать в другую программу или отправлять в виде сообщений электронной почты.

Запросы являются основным рабочим инструментом базы данных и могут выполнять множество различных функций. Самая распространенная функция запросов — извлечение определенных данных из таблиц. Данные, которые необходимо просмотреть, как правило, находятся в нескольких таблицах; запросы позволяют представить их в одной таблице. Кроме того, по-

сколько обычно не требуется просматривать все записи сразу, с помощью запросов можно, задав ряд условий, «отфильтровать» только нужные записи.

Макросы в приложении Access можно рассматривать как упрощенный язык программирования, который позволяет добавлять функциональные возможности в базу данных. Например, кнопке команды в форме можно назначить макрос, который будет запускаться при нажатии этой кнопки. Макрос содержит последовательность действий для выполнения определенной задачи, например для открытия отчета, выполнения запроса или закрытия базы данных. Большинство операций с базой данных, выполняемых вручную, можно автоматизировать с помощью макросов, которые позволяют существенно экономить время.

Модули, как и макросы, являются объектами, которые можно использовать для добавления функциональных возможностей в базу данных. В то время как макросы создаются в приложении Access путем выбора макрокоманд из списка, модули пишутся на языке программирования Visual Basic для приложений (VBA) (VBA (Visual Basic for Applications)). Версия макроязыка программирования Microsoft Visual Basic, используемая для программирования приложений для Microsoft Windows и поставляемая с некоторыми программами корпорации Майкрософт.).

Чтобы приступить к заполнению таблицы базы данных необходимо сначала дать названия полям и назначить им тип данных. Для этого необходимо перейти в «Режим»+ «Конструктор»:

В появившемся окне введите название таблицы и нажать ОК.

После этого Access выводит окно Конструктора таблицы, в котором задаются имена, типы и свойства полей для создаваемой таблицы. По умолчанию первое поле таблицы создается под именем КОД с типом данных Счетчик и предназначено быть ключевым полем (следует обратить внимание на значок КЛЮЧ слева от записи, который можно установить и для другой записи) (в данном столбце будет производиться нумерация всех введенных записей (строк) таблицы. Если это устраивает, то продолжаем заполнять имена полей и типов данных (выбирается из ниспадающего списка). Иначе переименовываем.

Имя поля не должно превышать 64 символа и в нем нельзя использовать символы пробел, точка, запятая, восклицательный знак.

Каждая строка в столбце Тип данных является полем со списком, элементами которого являются типы данных Access (таблица 1). Тип поля определяется характером вводимых в него данных.

Таблица 1. Типы данных MS Access 2007

Тип данных	Использование
Текстовый	Алфавитно-цифровые данные (до 255 символов)
Мемо	Алфавитно-цифровые данные-предложения, абзацы, тексты (до 64 000 символов)
Числовой	Различные числовые данные (целое, длинное целое, с плавающей точкой)
Дата\ Время	Дата или время в одном из предлагаемых Access форматов

Денежный	Денежные суммы, хранящиеся с 8 знаками в десятичной части. В целой части каждые три разряда разделяются запятой.
Счетчик	Уникальное длинное целое, создаваемое Access для каждой новой записи
Логический	Логические данные, имеющие значения Истина или Ложь
Объект OLE	Картинки, диаграммы и другие объекты OLE из приложений Windows
Гиперссылка	В полях этого типа хранятся гиперссылки, которые представляют собой путь к файлу на жестком диске, либо адрес в сетях Internet или Intranet.

Задания:

1 Запустить Microsoft Office Access 2007. Создать новую пустую базу данных под своим именем, сохранить ее в папке на диске: D:\, выбрав для нее группу

2 Создать таблицу 1 «Видеокарты» в режиме Таблица.

3 Описать структуру таблицы 2 «Учет программного обеспечения» в окне Конструктора. Осуществить ввод данных в нее.

4 С помощью шаблона создать таблицу 3 «Заказы», выбрав необходимый прототип таблицы.

5 По заданию преподавателя осуществлять модификации структуры таблицы «Видеокарты».

6 Пропровернуться с преподавателем, позволяющие выполнять порекомендованные данные внутри таблицы.

7 Пропровернуться с преподавателем по изменению внешнего вида таблицы.

Таблица 1 – Содержание таблицы «Видеокарты»

Но- мер	Модель	Произво- дитель	Частота, МГц	Объем памяти, МБ
1	DDR-3 Sapphire ATI RADEON X1650 Pro	Sapphire	550	256
2	DDR-2 Palit GeForce 5600GT Super	Palit	650	1024
3	DDR-3 Sapphire ATI RADEON HD4670	Sapphire	750	1024
4	DDR-2 ZOTAC GeForce 5600GT	ZOTAC	540	1024
5	DDR-3 ASUS/AsK EN6800GT/HTDP GeForce 6800GT	ASUS	600	512
6	DDR-2 ASUS EN4410GT MACH10M	ASUS	650	512

Таблица 2 – Содержание таблицы «Учет программного обеспечения»

Код	Программное обеспечение	Количество	Дата установки
1	ABRYV FireRadar 7.0	3	08.10.08
2	ASPI/asy-CDM	3	28.12.08
3	Microsoft Windows Vista Business RUS	1	19.02.09

4	Microsoft Windows Server 2003	2	06.12.08
5	Symantec Norton AntiVirus 2008	1	12.09.08
6	Microsoft Windows XP Professional	20	19.01.09
7	Microsoft Office 2007	15	15.12.08
8	Antivirus Касперского 2009 Рус.	20	09.01.09

Таблица 1.8 – Содержание таблицы «Заказы»

Код товара	Описание	Дата	Количество	Цена
1	LG L1734S.BN	15.10.08	50	3650
2	Lenovo F1400	06.12.08	7	3900
3	Samsung 3037RW HSPV	23.06.08	34	9800
4	Samsung SCX-4200	19.01.09	20	6200
5	Genius KB-110	03.07.08	15	820
6	Beluga 1730 SL	04.02.09	5	3370
7	Lenovo Deluxe 230 Y-SAPTE	14.09.08	27	160

8 В каждую созданную таблицу осуществить ввод данных, используя прайс-листы предприятий г. Курска, расположенные в свободном доступе в сети Интернет, добавив до 20 записей.

9 В отчете по практической работе дать ответы на тестовые задания и контрольные вопросы:

Тестовые задания и контрольные вопросы:

1 Что такое таблица в Access?

а) объект базы данных, в который добавляются элементы упорядоченны, структурированы на действия пользователей или служащие для ввода, отображения и изменения данных в поле;

б) объект базы данных, в котором данные хранятся в виде записей (строк) и полей (столбцов);

в) объект базы данных, предназначенный для вывода на печать данных, организованных и сформатированных в соответствии с требованиями пользователя;

г) набор условий, применяемых для выбора подмножества данных или для извлечения данных?

2 Набор всех хранящихся записей в одной таблице Access - это:

а) количество записей базы данных;

б) представление базы данных;

в) экранный файл;

г) экранный запрос;

д) физическая схема данных.

3 Укажите расширение файла базы данных, созданной в Access

а) acc.db;

б) db;

в) dbc;

г) doc;

д) dbf

4 Какая тип поля не поддерживает MS Access:

- а) логический;
- б) денежный;
- в) текстовый;
- г) текстовый примечаний;
- д) числовой?

5 Какие группы команд расположены на панели Главная?

- а) «Представления», «Буфер обмена», «Шрифты», «Таблицы»;
- б) «Сбор данных», «Буфер обмена», «Шрифты», «Записи»;
- в) «Представления», «Форматирование и тип данных», «Шрифты», «Записи»;
- г) «Представления», «Буфер обмена», «Шрифты», «Записи».

6 Каждое поле в базе данных может быть односно к одному из следующих типов:

- а) символьный, логический, числовой;
- б) логический, символьный, числовой, тип примечаний;
- в) логический, конкурентный, логический, форматный;
- г) логический, символьный, конкурентный, логический.

7 Поле логического типа содержит:

- а) величину, представляющую значение «истинно» или «ложно»;
- б) логические высказывания;
- в) суть логических рассуждений;
- г) логические знаки.

8 Для чего служит поле типа Memo:

- а) для хранения секретной информации;
- б) для реализации других полей;
- в) для хранения больших массивов данных в специальном файле;
- г) для графиков?

9 Нормализация – это операция:

- а) хранения информации;
- б) перемещения по базе данных;
- в) создания эффективной структуры базы данных;
- г) сортировка записей базы данных;
- д) выполнения действий пользователя.

10 Какими записи таблицы «Студенты» в Access может быть поле, содержащее следующие данные:

- а) номер группы;
- б) факультет;
- в) номер учебной книжки;
- г) изучаемая дисциплина;
- д) имя студента.

11 Для чего предназначена Access?

12 Перечислите основные функции Access.

13 Что содержит Лента Access?

14 Какие режимы работы имеет Access?

15 Какие элементы входят в состав любой базы данных Access?

16 Какие способы создания базы данных поддерживает Access?

	17 Что такое таблица?
	18 В чем заключается основное преимущество Ленты?
	19 Каким образом осуществляется создание пустой книги данных?
	20 Что такое «табличная книга»?
	21 Каким образом создается табличная книга Конструктора?
	22 Какие типы данных используются в Access?
	23 Что входит в описание структуры таблицы?
	24 Каким образом создается таблица с помощью шаблонов?
	25 Как создается таблица в режиме таблицы?
	26 Как осуществлять модификацию структуры таблицы?
	27 Как ввести данные в таблицу?
табл	28 Какие командные кнопки используются для перемещения по таблице?
	29 Каким образом можно выделить поле и запись?
	30 Как переместить данные из одного поля в другое?
	31 Как копировать информацию из нескольких полей?
	32 Как осуществляется изменение порядка следования полей?
	33 Как можно осуществить изменение ширины столбцов?
	34 С какой целью выполняется команда Скрыть столбцы?
	35 Каким образом можно сделать столбец выделенным?
табл	36 С помощью какой команды осуществляется форматирование столбца?
	37 Как осуществляется изменение параметров шрифта?
	38 Какие команды доступны в окне диалога «Формат таблицы»?
	39 Что включает в себя группа «Шрифт»?
	40 Как изменить нум. столбца?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 10

Тема: Поиск данных в MS Office Access 2007

Цель работы: освоить технологию поиска данных, сортировки и фильтрации данных в СУБД MS Office Access 2007.

Теоретический материал:

Поиск и замена данных в БД

Программа MS Access 2007 позволяет производить поиск записей, в которых значения определенного поля полностью или частично совпадают с некоторой величиной.

Примечание: для поиска используются следующие виды шаблонов:

Звездочка (*) — заменяет любую группу любых символов; может быть первым или последним символом в шаблоне.

Имея условием "Вас*", будут отобраны и "Василиса", и "Василий", и т.д.

Имея условием "*ова", можно отобрать все фамилии, оканчивающиеся на "ова" — Иванова, Петрова и т. д.

Знак вопроса (?) — заменяет любой один символ.

Если задать в качестве условия "М?ша", то будут отобраны и "Миша", и "Маша".

Знак номера (#) - заменяет любую одну цифру.

Задав условие "1#3", найдете "103", "113", "123"...

Поиск данных в БД с помощью фильтров

Гораздо больше возможностей для поиска данных в базах данных предоставляют фильтры. Фильтры позволяют отбирать записи, которые удовлетворяют заданным условиям. Условия отбора записей создаются с использованием операторов сравнения (=, >, < и т. д.).

I. Простые фильтры содержат условие отбора записей только для одного поля.

II. Сложные фильтры содержат условия для различных полей. В результате применения сложного фильтра будут отобраны только те записи, которые удовлетворяют всем условиям одновременно. Можно сказать, что условия в сложных фильтрах связаны между собой операцией логического умножения.

Задания:

1 Запустить MA Access, затем базу данных, содержащую таблицы «Видеокарты», «Учет программного обеспечения», «Заказы».

2 В таблице «Видеокарты» осуществить поиск тех моделей, которые имеют объем памяти 1024 Мб.

3 В таблице «Учет программного обеспечения» осуществить поиск программного обеспечения, имеющего название, начинающееся с «М», программного обеспечения, установленного в 2008 году.

4 В таблице «Учет программного обеспечения» произвести поиск и удалить строку с названием «23.12.08» на «15.01.09».

5 Осуществить в таблице «Видеокарты» сортировку по убыванию значений объема памяти; по возрастанию названия производителя и модели.

6 В таблице «Видеокарты» задать фильтр по выделенному значению «1024» в поле «Объем памяти».

7 В таблице «Видеокарты» вывести на экран все записи кроме тех, которые попадают в поле «Объем памяти» значение «1024».

8 В таблице «Видеокарты» вывести на экран с помощью обычного фильтра записи о видеокартах, которые содержат в названии модели «GeForce» и имеют объем памяти 1024 МБ.

9 В таблице «Видеокарты» с помощью обычного фильтра вывести на экран записи о видеокартах, цены ниже 3000.

10 В таблице «Учет программного обеспечения» с помощью расширенного фильтра осуществить выбор записей об операционных системах Microsoft и выполнить сортировку по убыванию даты установки.

11 С помощью мастера запросов сформировать из таблицы «Видеокарты» новую таблицу с полями «Модель», «Производитель», «Объем памяти» с отсортированными по возрастанию названий моделей. Удалить из созданного в пункте 2.2.11 запроса поле «Объем памяти».

12 Создать в конструкторе запросов в таблице «Учет программного обеспечения» на критерии, указанные в п. 2.2.3; а также изучить сведения о программном обеспечении, установленном с 01.12.2008 по 31.01.2009.

13 Перечислить перечень команд, используемых для выполнения практической работы, с указанием их назначения.

14 В отчете по практической работе дать ответы на тестовые задания и контрольные вопросы:

Тестовые задания и контрольные вопросы:

1 На какой вкладке Ленты находится группа «Найти»:

- а) Создание;
- б) Главы;
- в) Работа с базами данных;
- г) Режим таблицы?

2 Какой подстановочный символ соответствует любому текстовому символу?

- а) *;
- б) %;
- в) [?];
- г) #

3 Подстановочный символ # соответствует:

- а) любой цифре;
- б) любому текстовому символу;
- в) любому символу из диапазона;
- г) любому количеству букв, цифр или других символов.

4. Назовите фильтры данных:

- а) наделение записей по указанному критерию;
- б) распределение записей в определенном порядке;
- в) представление данных по указанному критерию;
- г) сортировка записей по указанному критерию;
- д) выбор записей по указанному критерию.

5. На какой вкладке Ленты находится команда «Сортировка»:

- а) Создание;
- б) Работа с базами данных;
- в) Главная;
- г) Режим таблицы?

6. Что является результатом команды «Применить фильтр»:

- а) таблицы, определенные заданным критерием;
- б) записей, определенных заданным критерием;
- в) поля, определенные заданным критерием;
- г) ячейки, определенные заданным критерием?

7. На какой вкладке Ленты находится команда «Создание запросов»:

- а) Создание;
- б) Работа с базами данных;
- в) Главная;
- г) Режим таблицы?

8. Что такое запрос в Access:

а) объект базы данных, в который добавляются элементы управления, регистрируемые на экране пользователя или служащие для сбора, отображения и изменения данных в поле;

б) объект базы данных, в котором данные хранятся в виде таблиц (строк и полей (столбцов));

в) объект базы данных, представляющий для вывода на печать данные, организованные в сформатированных и структурированных представлениях данных;

г) объект базы данных, позволяющий пользователю извлекать нужные данные по одной или нескольким таблицам?

9. Какая команда выполняется для формирования сложного критерия:

- а) Построитель;
- б) Формирователь;
- в) Запрос;
- г) Выбор?

10. Если в конструкции запроса дважды нажать «ОК» (*) и выбранной таблице, то это обозначит отображение запросов:

- а) всех полей из таблиц, которые он возвращает;
- б) всех записей, которые он возвращает;
- в) всех критериев, которые он возвращает;
- г) всех условий, которые он возвращает.

11. Какой командой осуществляется поиск записей?

12. Что задается в одно полевого «Поиск»?

13. Что используется для поиска данных, точное значение которых неизвестно?

14. Каким образом осуществляется поиск и замена данных?

15. Что означает в базе сортировка данных?

16. Назовите способы проведения сортировки.

17. Для чего в базе данных используется фильтр?

18. Как можно создать фильтр?

19. Какие виды фильтров известны в Access?

20. Как выполняется создание фильтра по заданному фрагменту?

21. Недостатки фильтра по заданному фрагменту.

22. Как создается обычный фильтр?

- 23 Как формируется выражение для критерия фильтра?
- 24 Что включает в себя тело оператора?
- 25 Как задать критерий с помощью расширенного фильтра?
- 26 Назовите отличие расширенного фильтра от обычного.
- 27 Что включает в себя тело диалога для установок критерия расширенного фильтра?
- 28 Что означает оператор?
- 29 Как создается запрос с помощью мастера?
- 30 Что включает в себя тело конструктора запросов?
- 31 Каким образом можно выбрать поля таблицы для сохранения запроса?
- 32 Как удалить лишние поля из запроса?
- 33 Как изменить порядок расположения полей в запросе?
- 34 Каким образом осуществляется запуск запроса?
- 35 Как сохранить запрос?
- 36 Как задать условия для выбора записей?
- 37 Каким образом строится условие при помощи логического оператора одного из полей?
- 38 Каким оператором выполняется при выборе записей по условию постоянного соединения таблиц?
- 39 Для чего используются подстановочные значения в запросах?
- 40 Что включает в себя тело построителя вычислений?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 11

Тема: Создание и использование форм для ввода и редактирования данных в MS Office Access 2007

Цель работы: освоить технологию создания пользовательских форм для ввода и редактирования данных на экране при работе с реляционными базами данных в СУБД MS Office Access 2007.

Теоретический материал:

В Access данные можно просмотреть непосредственно в таблицах. Однако это не всегда удобно, поскольку иногда необходимо вывести на экран все поля одной записи одновременно. Для облегчения работы пользователей создаются формы.

Форма – объект базы данных Access, в которой размещаются элементы управления, реагирующие на действия пользователей или служащие для ввода, отображения и изменения данных в базе.

Формы могут применяться для управления доступом к данным: с их помощью можно определить, какие данные или экраны данных будут отображаться. Например, некоторым пользователям достаточно видеть лишь несколько полей большой таблицы. Если предоставить им форму, содержащую только нужные им поля, это облегчит для них использование базы данных. Для автоматизации часто выполняемых действий в форму можно добавить кнопки и другие функциональные элементы. Формы можно рассматривать как окна, через которые пользователи могут просматривать и изменять базу данных. Различные пользовательские формы ускорят работу с базой данных, поскольку пользователям не придется искать то, что им нужно. Внешне пользовательная форма делает работу с базой данных более приятной и эффективной, кроме того, она может помочь в преобразовании простого ввода данных. В Microsoft Office Access 2007 предусмотрены новые средства, позволяющие быстро создавать формы, а также новые типы форм и функциональные возможности, благодаря которым базы данных станут еще более приятными. Также можно вывести формы на экран – это довольно ценное предложение в разрабатываемых отчетах, но в первую очередь формы предназначены для представления данных на экране.

Самым простым способом работы является использование форм, созданных при помощи инструмента «Формы». Чтобы создать такую форму, необходимо:

- открыть базу данных;
- в окне переключен базы данных перейти на вкладку «Таблицы»;
- установить указатель на таблицу, для которой создается форма;
- выполнить команду **Формы на Ленте на вкладке Создание** в группе «Формы».

Разделенная форма – это новая возможность в Microsoft Office Access 2007, позволяющая одновременно отображать данные в двух представлениях – в режиме формы и в режиме таблицы.

Эти два представления связаны с одним и тем же источником данных и всегда синхронизированы друг с другом. При выделении поля в одной части формы выделяется то же поле в другой части. Данные можно добавлять, изменять или удалять в каждой части формы (при условии, что источник данных допускает обновление, а параметры формы не запрещают такие действия). Работа с разделенной формой дает преимущества обеих типов формы в одной

форме.

Одним из способов создания форм, предоставляемых Ассекс, является Мастер форм. По сравнению с автоматически создаваемыми формами, создаваемыми с помощью мастера, более разнообразны по стилю оформления, содержат заданные поля. Кроме того, можно указать способ группировки и сортировки данных, а также включить в форму поля из нескольких таблиц или запросов, при условии, что заранее заданы отношения между этими таблицами и запросами.

Чтобы запустить Мастер форм, необходимо выполнить следующие действия: на панели Создания в группе «Формы» нажать Другие формы, а затем в списке выбрать пункт Мастер форм ().

После нажатия кнопки раскрытия списка Таблицы и Запросы необходимо выбрать из списка таблиц базы данных таблицу или запрос, для которой создается форма. При этом в списке Допустимые поля появятся перечни всех полей выбранной таблицы. Надо перенести из данного перечня в список Выбранные поля те поля, которые необходимо поместить в создаваемую форму. Для выбора полей из списка допустимых полей надо поместить их, а затем перенести в список выбранных полей, нажав мышью кнопку с одиночной стрелкой, изображенной внизу. Если надо исключить из списка, то нажимаем кнопку с двойной стрелкой. Таким же образом исключаются поля, выбранные из списка, пользуясь кнопками со стрелками, направленными влево.

Мастер форм позволяет выбирать формы, являющиеся либо таблицей одной таблицы, но и из нескольких связанных. В этом случае после выбора таблиц из первой таблицы необходимо выбрать их список таблиц, базой данных которой является таблица, к которой переносятся требуемые поля в список Выбранные поля.

Завершив формирование списка полей формы, необходимо нажать кнопку Далее, чтобы перейти в следующее окно, которое позволяет задать название и вид формы. Выбирая нужную иерархию, надо нажать кнопку Далее.

Затем мастер позволяет выбрать стиль формы. Из списка, содержащего варианты стилей, выбирается стиль, который в наибольшей степени отвечает требованиям. После этого необходимо нажать кнопку Далее. На экране отобразится предварительный просмотр. Необходимо нажать в нем кнопку Изменить форму. Затем указывается вариант дальнейшей работы установкой одной из опций:

- открытие формы для просмотра или ввода данных (сохраняет созданную форму и открывает ее для просмотра или ввода данных);
- изменение макета формы (сохраняет выбранную форму и открывает ее в конструкторе форм для модификации).

После установкой требуемых опций необходимо нажать Готово для завершения создания формы и появления Мастера.

Автоматически создаваемые формы и мастер форм позволяют быстро создать экранную форму для ввода и просмотра данных. Но для таблиц, имеющих много полей, эти формы не позволяют разместить на экране одновременно все поля таблицы. С таблицей даже работать, если разгруппировать все поля в таблицы так, чтобы можно было просматривать их одновременно. Для создания сложных и более удобных для конечного пользователя форм средства простых форм и мастера недостаточны. Создавать формы любой степени сложности можно с помощью конструктора форм.

Любая форма в Access состоит из объектов формы, которые имеют характерные для них свойства. Для каждого объекта можно определить действия, выполняемые при наступлении определенных событий. Процесс создания формы состоит из размещения объектов и форм и определения для них свойств, связанных с ними событий и выполняемых действий.

Создать форму можно, выполнив следующие действия:

- открыть базу данных;
- в области переноски базы данных нажать на табличку, на основе которой необходимо построить форму;
- нажать кнопку Конструктор форм на Ленте в группе «Формы» на вкладке Создание.

Все объекты Access характеризуются свойствами, которые можно настроить в соответствии с требованиями. Кроме того, для каждого объекта существуют определенные события, которые выполняются при наступлении связанных с ними действий.





Для получения доступа к свойствам и событиям объекта необходимо выделить нужный объект и выполнить одно из следующих действий: выбрать команду Свойства из контекстного меню или нажать кнопку Страница свойств на Ленте в группе «Формы».

В результате на экране появится окно диалога со свойствами и событиями выбранного объекта (рисунком 3.6). В верхней части окна диалога показаны ярлычки с перечнем сгруппированных по типам свойств и событий:


- Макет (содержит свойства объекта, связанные с его оформлением);
- Данные (содержит свойства объекта, связанные с источниками данных);
- События (содержит список всех событий объекта);
- Другие (в данную вкладку собраны все свойства, не вошедшие во вкладки Данные, Макет и События);
- Все (содержит список всех свойств и событий формы в алфавитном порядке).

При размещении объекта в форме устанавливаются принятые по умолчанию значения свойств объекта. Для того, чтобы изменить стандартные установки для свойства какого-либо из объектов, необходимо выбрать объект и открыть

окно его свойств. Возможно следующие типы значений свойств:

- свойства математического редактирования (числа, даты, время, текст, а также можно ввести с клавиатурой требуемые значения);
- возможные два или более различных вариантов значений свойства (рядом с полем коррекции появляется кнопка раскрытия списка );
- возможен выбор свойств с помощью выпадающего списка (рядом с полем коррекции появляется кнопка раскрытия списка параметров );
- создание выражения (рядом с полем коррекции свойства появляется кнопка раскрытия списка, а рядом с ней кнопка открытия окна процедур, макросов или построителя выражений в зависимости от выбранного значения списка  ).

В процессе создания формы можно перемещать, удалять или изменять размеры объектов, а также изменять их свойства.

Для удаления объектов прежде всего нужно выделить требуемый объект. Для выделения одного объекта достаточно установить указатель мыши на объект и нажать кнопку мыши. Для выделения одновременно нескольких объектов можно воспользоваться любым из следующих способов: нажать клавишу Shift и, удерживая ее, нажать последний, нужный для выделения объект или выбрать инструмент «Выбор объектов» на панели элементов (); установить указатель мыши на объект и, не отпуская кнопки мыши, переместить рамку выделения так, чтобы внутри нее оказались выделяемые объекты.

Для снятия выделения со всех объектов установить указатель мыши вне выделенных объектов и нажать кнопку мыши. Если необходимо снять выделение с отдельных объектов, то, удерживая нажатой клавишу Shift, выделить мышью объекты, с которых требуется снять выделение. Выделенный объект имеет маркеры выделения в виде квадратиков, расположенные по углам и серединам сторон. Для изменения размера объекта надо переместить один из маркеров до достижения нужного размера.

Если требуется установить точные размеры объекта, можно воспользоваться альтернативным способом. Для этого необходимо открыть окно свойств объекта и установить требуемые значения высоты и ширины в полях ввода значащих свойств Ширина и Высота.

Для изменения масштаба объекта необходимо выделить его, установить указатель мыши на объект выбранного объекта, появиться маркеры из стрелками. Теперь, удерживая кнопку мыши правой, переместить объект к требуемому месту.

После ввода в рисунок являются связанными объектами, то есть они перемещаются вместе. Для перемещения связанных объектов отдельно друг от друга понадобится нажать перемещение (большой квадрат в верхнем левом углу). Когда подведете указатель мыши к метке перемещения, как поле ввода, так и связанной с ним надписи, он превратится в маркеры со стрелками. Если теперь нажать мышью и, удерживая кнопку правой, можно переместить объект, при этом остальные объекты останутся на месте.

Если требуется удалить объект, необходимо выделить его, затем нажать клавишу Del.

Для увеличения внешнего вида формы применяется выделение выбранных объектов относительно сетки формы или друг друга, а также установка одинаковых размеров объектов. Для этих целей используется команда на панели Упорядочить. Блок панели содержит набор функций выравнивания и наметки размеров объектов. Чтобы сделать все объекты одинаковыми по размеру, выделите группу объектов, нажима на строку «Выровнять по размеру» по ширине; по высоте; по размеру; по самому большому; по самому широкому; по самому узкому.

А также обеспечивает средствами точного размещения объектов при построении сетки. Для этого необходимо выполнить команду Привязать из группы «Объект» панели «Упорядочить». Затем переместить объект при помощи мыши. Во время перемещения контур объекта отображается на горизонтальной и вертикальной линиях.

Во время ввода данных переход от одного объекта к другому при нажатии кнопки Таб осуществляется к соответствию с заданным в заданной форме порядком объектов. Для того, чтобы переопределить порядок объектов объектов и форма, надо выбрать команду Переходы из группы «Меню элемента управления». На экране откроется одно диалого «Последовательность переходов». Выделить в окне список темой перемещаемый объект, нажать кнопку, расположенную слева от названия объекта. Затем нажать кнопку еще раз и переместить ее в требуемое место в списке объектов.

Кнопки используются в формах для выполнения определенного действия или ряда действий. Например, можно создать в форме кнопку, открывающую другое форму, или создать набор кнопок для перемещения по записям таблицы, если не устраивают стандартные средства перемещения, предусмотренные в форме. Для того, чтобы кнопки выполняли какое-либо действие, необходимо создать макрос или процедуру обработки события и связать их со свойством кнопки Нажатие кнопки.

В Access предусмотрено создание более 30 разных кнопок, что значительно облегчает работу пользователя, избавляя его от необходимости самостоятельно разрабатывать макросы, достаточно лишь воспользоваться Мастером по созданию кнопок. Для добавление в форму командной кнопки необходимо установить режим нажатия кнопки на Ленте в группе «Элементы управления» и нажать кнопку Кнопка. Затем установить мышь в то место формы, где будет размещаться кнопка, определить тем самым ее размеры и местонахождение.

Открывающееся первое диалоговое окно Мастера позволяет выбрать категорию, то есть тип команд, которую должна выполнять кнопка. Содержимое в окне команды для нажатия по записям совпадают с теми действиями, которые осуществляются по интерфейсу Access. При перемещении по списку Категории список Действия обновляется. Окно Образов показывает изображение кнопки со стандартной иконкой, которая представляет данную команду. После выбора нужного изображения из списка Действия нажать кнопку Далее. Второе диалоговое окно позволяет разными способами заменить изображение на кнопку. Можно щелкнуть на переключателе Рисунок и воспользоваться стандартными иконками Access или использовать рисунокый файл в виде иконки кнопки. Кроме того, есть возможность выбора переключателя Текст, чтобы кнопка иконки в форме присутствовал текст. При использовании стандартных кнопок надо выбрать флажок Показать все рисунки, чтобы отобразить все имеющиеся стандартные кнопки, или отметить установку флажка, чтобы вывести только те кнопки, которые предлагаются программой для этой кнопки.

При использовании рисунков в rasterных файлах необходимо щелкнуть по переключателю Рисунок, нажать на кнопку Обзор, чтобы вывести на экран диалоговое окно, которое позволяет выбрать любой файл формата BMP и разместить иконку элемента управления. При использовании текста необходимо выбрать переключатель Текст, затем ввести нужный текст в поле справа.

Последнее диалоговое окно позволяет дать имя и завершить создание командной кнопки.

В качестве примера рассмотрим создание кнопки, позволяющей перейти по нажатию на эту кнопку в другой экран:

- открыть разрабатываемую форму в режиме конструктора;
- перейти в нижнюю часть формы и курсором мыши выделить Область данных на два столбца ширины, чтобы на эту свободную часть формы поместить кнопку;
- нажать инструмент Кнопка, чтобы поместить кнопку на форму, после этого откроется окно Мастера кнопки;
- в первом диалоговом окне выбрать раздел Переходы по записям в левом окне формы Категория, в правом окне формы Действия выбрать команду Первая запись, нажать кнопку Далее;
- во втором диалоговом окне установить переключатель на рисунок и выбрать Первая запись из меню, нажать кнопку Далее;
- в следующем окне задать наименование кнопки, в третьем окне ввести наименование «Первая», нажать кнопку Готово.

Задания:

1. Создать таблицу «Видеокарты» в соответствии с таблицей 1.

Таблица 1 Содержание таблицы «Видеокарты»

Номер	Модель	Производитель	Частота чипа, MHz	Объем памяти, MB	Наличие на складе
1	DDR 3 Sapphire ATI R4-DEON X1650 Pro	Sapphire	590	256	Да
2	DDR-2 Palit GeForce 3600GT	Palit	650	1024	Нет
3	DDR-3 Sapphire ATI R4-DEON HD4670	Sapphire	750	1024	Да
4	DDR-2 ZOTAC GeForce 8800GT	ZOTAC	540	1024	Да
5	DDR 3 ASUS/Star EN9400GT GeForce 8800GT	ASUS	600	512	Нет
6	DDR-2 ASUS EN9400GT MAGNOM	ASUS	650	512	Да
7	DDR 2 ASUS EN9600GT MG	ASUS	650	512	Нет
8	DDR-2 ZOTAC GeForce 8800GT	ZOTAC	540	1024	Да

2. Создать и сохранить форму для таблиц «Видеокарты» и «Заказы» с помощью инструмента «Форма».
3. Создать и сохранить разделенную форму для таблицы «Видеокарты».
4. Создать и сохранить экранную форму с помощью мастера для таблиц «Видеокарты», «Учет программного обеспечения».

4.5 Создать форму с помощью конструктора. Сменить тему с его именем, открыть как рабочую область. Настроить форму для таблицы «Видеокарты», разместить текстовую информацию, поля ввода, кнопки управления, список поля со списком «Производители» и «Объем памяти», флажок поля «Настройка склада» в соответствии с рисунком 1

6 Создать таблицу «Сведения о складе» (таблица 2)

7 Создать форму для ввода данных, как показано на рисунке 2. Для предоставления значений поля «Объем памяти» использовать выражения. Значения поля «Вид оплаты» задавать с помощью переключателей.

8 Протестировать работу всех созданных форм

Рисунок 1 - Экранная форма «Видеокарты»

Таблица 2 Содержание таблицы «Сведения о заказе»

Код	Тип товара	Модель	Код товара	Дата покупки	Кол-во шт	Цена	Инд. оплаты
1	Монитор	LG L1714S-BN	15000	15.11.08	50	3450	Бухгалтер
2	Принтер	Lexmark P1000	12588	05.04.08	7	3900	Од. закуп.
3	Монитор	Samsung 16928W R5FY	37000	23.08.08	24	8400	Бухгалтерия
4	Принтер	Samsung SCX-4200	43000	24.03.08	30	6300	Бухгалтер
5	Компьютер	Gigabyte CR330	5000	03.07.08	15	550	Бухгалтерия
6	Монитор	Benq 1750 SL	25000	04.02.08	2	3500	Од. закуп.
7	Компьютер	Logitech Deluxe 350 V. 5AFT6	7000	04.04.08	37	740	Бухгалтерия

Рисунок 3 Экранная форма «Сведения о заказах»

9 В отчете по практической работе дать ответы на тестовые задания и контрольные вопросы:

Тестовые задания и контрольные вопросы:

1. Форма в Access - это:

а) объект базы данных, в который добавляются элементы управления, реагирующие на действия пользователей или служащие для ввода, отображения и изменения данных в базе;

б) объект базы данных, в котором данные хранятся в виде записей (строк) и полей (столбцов);

в) объект базы данных, предназначенный для наглядного показа данных, предназначенных к официальному использованию с требованиями пользователя;

г) набор условий, применяемых для отбора подмножества данных или для сортировки данных.

2. Что является достоинством формы, созданной при помощи инструмента «Формы»:

а) легкий способ создания формы;

б) позволяет редактировать данные;

в) позволяет вводить данные;

г) позволяет изменить данные?

3. Что представляет собой разнесенная форма:

а) отображение данных в режиме формы;

б) одновременное отображение данных в режиме формы и таблицы;

в) отображение данных в режиме таблицы;

г) отображение данных в режиме формы с возможностью перехода по записям;

д) отображение данных в режиме таблицы с возможностью перехода по записям?

4. Назовите правильную последовательность этапов создания формы с помощью Мастера формы:

а) изменение внешнего вида формы, применение требуемого стиля, выбор полей для формы, задание имени формы, выбор действия;

б) выбор полей для формы, применение требуемого стиля, задание имени формы, изменение внешнего вида формы, выбор действия;

в) применение требуемого стиля, выбор полей для формы, изменение внешнего вида формы, задание имени формы, выбор действия;

г) выбор полей для формы, изменение внешнего вида формы, применение требуемого стиля, задание имени формы, выбор действия;

д) выбор действия, выбор полей для формы, изменение внешнего вида формы, применение требуемого стиля, задание имени формы.

5. Какой внешний вид формы нельзя выбрать при создании формы с помощью Мастера:

а) в один столбец;

б) табличный;

в) в одну строку;

г) вертикальный;

д) табличный?

- 6 На вкладке «Управление элементами» группы:
- а) «Шрифты», «Макет элемента управления», «Элементы управления», «Размеры», «Положения»;
 - б) «Автоформаты», «Макет элемента управления», «Выравнивание элементов», «Размеры», «Сетки»;
 - в) «Шрифты», «Элементы управления», «Выравнивание элементов», «Размеры», «Положения»;
 - г) «Автоформаты», «Макет элемента управления», «Выравнивание элементов», «Сетки», «Положения»;
 - д) «Автоформаты», «Макет элемента управления», «Выравнивание элементов», «Размеры», «Положения».

7 На какой вкладке Ленты находится команда Конструктор форм?

- а) Ссылки
- б) Работа с базой данных
- в) Таблицы
- г) Работа с таблицей

8 Какой элемент, помещаемый на форму, позволяет реализовать пункт:

- а) надпись;
- б) поле;
- в) выключатель;
- г) флажок;
- д) кнопка?

9 Какой элемент, помещаемый на форму, позволяет задать значение логического типа

- а) надпись;
- б) поле;
- в) выключатель;
- г) флажок;
- д) кнопка?

10 Какой элемент, помещаемый на форму, позволяет осуществить различные действия

- а) надпись;
- б) поле;
- в) выключатель;
- г) флажок;
- д) кнопка?

11 Для чего используются привязанные формы?

12 Какие команды используются для создания форм в Access?

13 Какие команды предназначены для создания привязанных форм?

14 Что представляет собой форма, созданная с помощью инструмента «Форма»?

15 Как переключить режим формы в режиме таблиц?

16 Каким образом создается форма с помощью мастера?

17 Что включает окно конструктора форм?

18 Для каких случаев используется конструктор форм?

19 Какие области имеет окно конструктора форм?

20 Как изменяется размер области в форме?

21 Какие группы инструментов находятся на вкладке Конструктор?

22 Какие инструменты находятся в группе «Элементы управления»?

23 Как получить доступ к свойствам объекта?

24 Что включает свойства объекта?

- 25 Какие возможны действия над объектами?
- 26 Как можно выделить объект?
- 27 Как можно изменить размер объекта?
- 28 Как изменяется местоположение объекта?
- 29 Какие объекты можно подмещать сквадным объектам?
- 30 Как удалить объект?
- 31 Как изменить размер объекта?
- 32 Как изменить порядок обхода объектов в форме?
- 33 Какие процедуры выполняются при создании формы?
- 34 Какие действия выполняются при настройке формы?
- 35 Какие действия выполняются при размещении текста в форме?
- 36 Какие объекты доступны для текста?
- 37 Как добавить в форму панель ввода?
- 38 Какие объекты можно связать поле с выражением?
- 39 Для чего используются в форме кнопки?
- 40 Как добавить в форму командную кнопку?
- 41 Что задается при работе мастера по созданию кнопок?
- 42 Для чего используются в формах линии и прямоугольники?
- 43 Для чего используется раскрывающийся список?
- 44 В чем заключается отличие «выпадающего списка» от «списочного»?
- 45 Для чего используется «флажок»?
- 46 Какой тип должна иметь панель, для которого используется флажок?
- 47 Для чего используется «группа переключателей»?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 12

Тема: Создание отчетов в MS Office Access 2007

Цель работы: освоить технологию создания и использования отчетов в СУБД MS Office Access 2007, построенных для реляционной базы данных.

Теоретический материал:

Access предоставляет в распоряжение пользователей средства для создания отчета. Отчет – объект базы данных Microsoft Access, предназначенный для вывода на печать данных, организованных и отформатированных в соответствии с требованиями пользователя.

При создании отчета можно воспользоваться стандартными средствами, ускоряющими процесс создания отчета или разработать специальный формат с помощью конструктора отчетов. Конструктор отчетов позволяет создавать отчеты, как в табличном виде, так и в свободной форме.

Табличный отчет представляет собой напечатанную таблицу, в которой данные упорядочены по столбцам и строкам. Каждый из столбцов содержит поле исходной таблицы или вычисляемое поле, а строка представляет собой запись. Табличный отчет позволяет печатать данные из таблицы в наиболее прямом и логичном виде. Однако они не пригодны в тех случаях, когда для исходной таблицы должны использоваться специально отведенных для печати поля (например, даты, часы, минуты).

Отчеты в свободной форме позволяют встраивать графические, специализированные табличные объекты. При получении отчета в свободной форме можно использовать стандартный формат, автоматически создаваемый Access для каждой таблицы. В этом формате поле исходной таблицы расположено вертикально. Однако с помощью конструктора отчетов можно разработать специальный формат, где поле исходной таблицы расположено в требуемых местах отчета.

В Access используются следующие средства для создания отчета:

- а) Отчет, позволяющий автоматически создать отчет с полем, расположенным в одной или нескольких столбцах;
- б) Мастер отчетов, позволяющий создать простейший отчет по одним выбранным полям;
- в) Конструктор отчетов, в котором можно самостоятельно разработать табличные отчеты с таблицами соотношений;
- г) Пустой отчет, позволяющий самостоятельно создавать текст и макеты, упрощающие и дорабатывать форму отчета;
- е) Наклейки, позволяющие создать отчет для почтовых адресов или других этикеток.

Для создания самого простого отчета в Access необходимо выполнить следующие действия:

- открыть окно базы данных;
- убедиться, что в области переходов выбрана таблица;

- напечатать команду Отчет в группе «Отчеты», на Ленте во вкладке Создание.

На экране появится тотный к редактированию отчет, в который встроены все поля таблицы. Их названия располагаются горизонтально и тем цветом, в котором они находятся в таблице. Снизу под названием каждого поля отображается его значение в таблице.

Расположение полей и записей в аккомодировке табличных отчетов подходит не для всех случаев, в частности, при их большом количестве. Однако отчеты могут пригодиться, когда создаются на основе запросов, выводящих на экран только нужные поля. Можно также использовать запрос для определения тех записей и полей их сортировки, которые будут включены в отчет.

Чтобы запустить мастер отчетов, необходимо нажать на кнопку Мастер отчетов в группе «Отчеты» на вкладке Создание.

Access рекомендует базовый набор элементов для просмотра на экране созданного отчета. Для просмотра отчета необходимо нажать кнопку Office



, навести стрелку рядом с пунктом Печать и нажать кнопку Предварительный просмотр

При просмотре отчета можно использовать кнопки панели инструментов и кнопки перемещения по страницам, расположенным в нижней части окна просмотра.

Режим конструктора позволяет более подробно изучить структуру отчета. Показатель может просматривать записки и примечания для поля, определенной страницы и групп. В этом режиме отчет не выводится, поэтому во время работы невозможно просматривать базовые данные. Однако некоторые задачи удобнее выполнять в режиме конструктора, а не макета.

В отчете можно добавлять различные элементы управления, такие как надписи, рисунки, линии и прямоугольники. Можно изменять источник элемента управления «Поиск» непосредственно в самом поле, без использования окна свойств. Можно также изменить определенные свойства, недоступные в режиме макета.

Для открытия окна конструктора отчетов необходимо открыть на Ленте вкладку Создание, нажать кнопку Конструктор отчетов в группе «Отчеты».

Задания:

- 1 Запустить Access, открыть свою базу данных.
- 2 Добавить в таблицу «Видеокарты», созданную в предыдущей лабораторной работе, новые поля «Цена» и «Дата поступления», заполнить поля значениями.
- 3 Создать отчет с помощью команды Отчет для таблицы «Видеокарты».

4 Создать отчет с помощью мастера для таблицы «Видеокарты», группировку задать по полю «Производитель» и «Дата поступления», задать интерес группировки по второму полю «Дата поступления», внутри групп идентифицировать по возрастанию значений в поле «Модель». В отчет не включать поле «Номер», вместо этого вывести по полю «Цена».

5 Создать отчет в структуре отчетов по заданию пункта 4. Сформировать заголовки отчета, групп, разместить дату печати отчета в нижнем отделе, пронумеровать значения в группе.

6 Перечислить перечень команд, используемых для выполнения практической работы, с указанием их назначения.

7 В отчете по практической работе дать ответы на тестовые задания и контрольные вопросы:

Тестовые задания и контрольные вопросы:

1 Отчет – объект базы данных, основные возможности которого:

- а) описание и вывод на печать документов на основе данных базы;
- б) ввод, отображение и изменение данных в полях;
- в) указание, какие данные из базы данных будут отображаться;
- г) создание и редактирование базы данных;
- д) структурирование описания автоматических выполняемых действий.

2 На какой вкладке Ленты находится команда Конструктор отчетов:


- а) Главная;
- б) Работа с базами данных;
- в) Создание;
- г) Режим таблицы?

3 Команда Мастер отчетов позволяет:

- а) автоматически издать отчет с полями, расположенными в один или несколько столбцов;
- б) создать пространственный отчет на основе выбранных полей;
- в) самостоятельно разработать собственную форму с заданными свойствами;
- г) самостоятельно вставлять поля и элементы управления и дорабатывать форму отчета.

4 Укажите правильную последовательность этапов создания отчета мастером отчетов:

- а) выбор полей, добавление уровня группировки, задание порядка сортировки записей, выбор варианта дальнейшей работы;
- б) выбор полей, добавление уровня группировки, задание порядка сортировки записей, выбор варианта дальнейшей работы, задание имени отчета;
- в) выбор полей, задание порядка сортировки записей, задание имени отчета, добавление уровня группировки, выбор варианта дальнейшей работы;
- г) выбор полей, задание порядка сортировки записей, добавление уровня группировки, задание имени отчета, выбор варианта дальнейшей работы;
- д) выбор полей, добавление уровня группировки, задание порядка сортировки записей, задание имени отчета, выбор варианта дальнейшей работы.

6. Кнопка  в окне диалогов мастера печати при отображении группировки предназначена для:

- а) добавления поля в рамку группировки;
- б) удаления поля из рамки группировки;
- в) повышения уровня группировки, выделенного в рамке группировки поля;
- г) понижения уровня группировки, выделенного в рамке группировки поля.

7. На какой вкладке Ленты находится команда Группировка и сортировка:

- а) Главная;
- б) Работа с базой данных;
- в) Сводные;
- г) Расчеты таблиц;
- д) Конструктор?

8. Для чего применяется разбивание данных на группы:

- а) для нумерации страниц;
- б) для объединения различных данных;
- в) для автоматического расширения информации в отчете;
- г) для группировки данных?

9. На какой вкладке Ленты находится команда Столбцы, используемая для изменения расположения информации в отчете:

- а) Экспертиза;
- б) Работа с базой данных;
- в) Сводные;
- г) Параметры страницы;
- д) Конструктор?

10. Какая функция позволяет оставить текущую дату печати отчета:

- а) Now;
- б) Time;
- в) Date;
- г) Day?

11. Укажите иконку команды Панора страница:

- а) ; б) ; в) ; г) ; д) .

12. Что понимается под отчетом в MS Access?

13. Что представляет собой отчет, созданный командой Отчет?

14. Какие в Access используются средства для создания отчетов?

15. Каким образом создается отчет с помощью мастера?

16. Как осуществляется предварительный просмотр отчета?

17. Что включает в себя конструктор отчета?

18. Как добавлять в окно конструктора элементов печати, если эти элементы не были сформированы?

19. Что является основными элементами отчета?

20. Как разместить дату печати отчета в заголовке отчета?

- .20 Как в конструкторе отчета осуществлять группировку данных?
- .21 Как указать сортировку или группировку в отчете?
- .22 Каким образом можно поместить на каждой странице отчета ее номер?
- .23 Какие действия выполняются при необходимости нумерации записи в группе?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 13

Тема: Реляционные базы данных в MS Office Access 2007

Цель работы: формирование навыков по созданию и работе с реляционной базой данных, состоящей из нескольких таблиц, связанных друг с другом с помощью общих ключевых полей, в MS Access 2007.

Теоретический материал:

Access является системой управления реляционными базами данных. Реляционные базы данных в настоящее время наиболее распространены и фактически являются промышленным стандартом. Единственной трудностью в реляционной базе данных информации является таблица. Каждая таблица представляет собой совокупность строк и столбцов, где строки (записи) представляют конкретному объекту, объекту или явлению, а столбцы – атрибутам (признакам, характеристикам, параметрам) объекта, события, явления.

Между отдельными таблицами базы данных могут существовать связи.

Установление связи между таблицами обеспечивает следующее:

- повышение достоверности хранимой в базе данных информации, так как многие СУБД автоматически выполняют контроль целостности данных, вводимых в базу, в соответствии с установленными связями;

- обеспечение доступа к данным при выполнении таких операций, как поиск, просмотр, редактирование, выборка и подготовка отчетов, при этом уменьшается количество запросов обращения к таблицам данных и число выполняемых запросов на них.

В каждой таблице базы данных может существовать первичный ключ – поле или набор полей, однозначно идентифицирующий запись. Значение первичного ключа в таблице базы данных должно быть уникальным, т.е. в таблице не должно существовать двух или более записей с одинаковым значением первичного ключа. Первичные ключи обеспечивают установление связи между таблицами.

В базе данных возможно четыре типа отношений между таблицами: один-к-одному, один-ко-многим, много-к-одному, много-ко-многим.

При установлении связи «один-к-одному» каждая запись в первой таблице может иметь не более одной связанный запись во второй таблице и наоборот. Отношения этого типа используются нечасто, поскольку обычно сведения, связанные таким образом, хранятся в одной таблице. Отношение «один-к-одному» используется для разделения таблицы, содержащей много полей, с целью выделения части таблицы по соображениям безопасности, а также с целью сохранения сведений, относящихся к подмножеству записей в главной таблице. После определения такого отношения у обеих таблиц должно быть общее поле.

Отношение «один-ко-многим» означает, что одна запись из родительской таблицы может соответствовать нескольким записям в дочерней таблице. Данный тип связи является самым распространенным для реляционных баз данных. Связанные отношениями таблицы взаимодействуют по принципу «главная – подчиненная». Главную таблицу часто называют родительской, а

потомственному - дочерний. Одна и та же таблица может быть главной по отношению к одной таблице базы данных и дочерней по отношению к другой. Например, связь «один-ко-многим» имеется между таблицами «Читательский билет студента» и «Выдана книга». Связь между таблицами осуществляется на основании значений соответствующих полей «Код читательского билета».

Отношение «оного-к-одному» аналогичен типу «один-ко-многим». Тип отношения зависит от выбранной точки зрения. Например, если рассматривать отношение между таблицами «Уроки» и «Конспекты», то получится отношение «оного-к-одному».

Отношение «многие-ко-многим» возникает между таблицами тех случаев, когда:

- одна запись из первой таблицы может быть связана более, чем с одной записью из второй таблицы;
- одна запись из таблицы может быть связана более, чем с одной записью из первой таблицы.

Для реализации базы данных проектирование логической структуры заключается в том, чтобы разбить всю информацию по таблицам (или в терминах реляционной модели - по отношениям), а также определить состав полей (в терминах реляционной модели - атрибутов) для каждой из этих таблиц.


Процесс проектирования является творческим, в немалой степени зависит от опыта и интуиции разработчика. Основными целями при разработке эффективной структуры данных являются:

- обеспечение быстрого доступа к данным в таблицах;
- исключение ненужного повторения данных, которое может являться причиной ошибок при вводе и нерационального использования дискового пространства компьютера;
- обеспечение целостности данных таким образом, чтобы при изменении одних объектов автоматически происходило соответствующее изменение связанных с ними объектов.

Процесс построения эффективной структуры данных для уменьшения избыточности информации в базе данных называется нормализацией. В теории нормализации баз данных разработаны достаточно формализованные подходы по разбиванию данных, обладающих сложной структурой, среди нескольких таблиц. Теория нормализации оперирует с пятью нормальными формами таблиц. Избыточность информации уменьшается от первой до пятой нормальной формы. Поэтому каждая последующая нормальная форма должна удовлетворять требованиям предыдущей формы и некоторым дополнительным условиям. При проектировании проектировании баз данных четвертая и пятая формы, как правило, не используются, поэтому ограничимся рассмотрением первых трех нормальных форм.

В Access можно устанавливать постоянные связи между таблицами, которые будут поддерживаться при создании форм, отчетов и запросов. Устанавливая связи между двумя таблицами, необходимо выбрать поле, которое содержит одну и ту же информацию. Поля, с помощью которых устанавливается связь между таблицами, могут иметь различные имена, но удобнее использовать совпадающие имена.

Порядок переименования связей между таблицами или их удаления необходимо закрыть все открытые таблицы. Создание связей между таблицами в Access осуществляется в окне диалогов «Схема данных», которое открывается по

команде: **Самый паннелик**  **находясь** на Ленте в группе «Показывать или скрывать панель задач» **Работа с базой данных**. Если в базе данных не заданы связи, автоматически открывается диалоговое окно «Добавление таблицы». Если оно не отображается, то на панели Конструктор в группе «Связи» нажать кнопку **Добавить таблицу**. В аналогичном окне «Добавление таблицы» отображаются все таблицы и запросы базы данных. Чтобы просмотреть только таблицы, выбрать пункт **Таблицы**. Чтобы просмотреть только запросы, выбрать пункт **Запросы**. Чтобы просмотреть и таблицы, и запросы, выбрать пункт **Все**.

В окне диалогов «Схема данных» имеется возможность не только создавать связи между таблицами, но и выполнять следующие действия:

- изменить структуру таблицы;
- изменить существующую связь;
- удалить связь;
- удалить таблицу из окна диалогов «Схема данных»;
- вывести на экран все существующие связи или связи только для конкретной таблицы;
- определить связи для запросов, не задавая условия целостности данных.

Если при создании связи в окне диалогов «Схема данных» возникает необходимость изменения структуры таблицы, то можно не покидая окно диалогов внести нужные изменения:

- установить указатель мыши на идентифицируемую таблицу;
- нажать первую кнопку мыши и выбрать из контекстного меню команду **Конструктор таблицы**;
- внести в структуру таблицы необходимые изменения;
- нажатием кнопки **Изменения**, нажать кнопку закрытия окна в строке заголовка, в ответ на запрос о сохранении изменений выбрать **Да** для сохранения изменений и возвращении в окно диалогов «Схема данных».

Для изменения существующей связи выполняется следующая последовательность действий:

- находясь в окне базы данных, нажать кнопку **Изменить связи** на Ленте в группе «Связи»;
- установить указатель на линию связи, которую требуется изменить, и дважды нажать кнопку мыши;
- в открывшемся окне диалогов «Изменение связей» внести нужные изменения и нажать кнопку **ОК**.

Для удаления связи необходимо открыть окно диалогов «Схема данных», установить указатель на удаляемую линию связи, выделить ее, нажать кнопку мыши, а затем нажать клавишу **Delete**. Когда Access предложит подтвердить удаление связи, нажать кнопку **Да**.

Для удаления таблицы из макета схемы данных необходимо открыть окно диалогов «Схема данных», выбрать удаляемую таблицу и нажать клавишу **Delete**. Таблица будет удалена из макета схемы данных вместе с определенными для нее связями.

В окне диалога «Наложение связей» возможно определение условий целостности данных. Целостность данных является одним из самых важных требований, предъявляемых к базе данных. Условием целостности данных является набор правил, используемых для поддержания связей между записями в связанных таблицах. Эти правила делают невозможным случайное удаление или изменение связанных данных.

Условия целостности данных выполняются при следующих условиях:

- связанное поле главной таблицы является ключевым полем или имеет уникальный индекс;
- связанные поля имеют один тип данных;
- обе таблицы принадлежат одной базе данных Access.

Невозможно определить условия целостности данных для присоединенных таблиц из баз данных других форматов.

При выборе данных из таблиц часто используется многотабличное выражение, поскольку информация в реляционных базах данных содержится не в отдельных таблицах, а в совокупности связанных таблиц. Запросы в реляционных базах данных выполняются объединенными запросами. Языком Access язык запросов по образцу SQL позволяет задавать многотабличные запросы к базе данных путем заполнения предлагаемой СУБД запросной формы. При выполнении запроса связанных таблиц добавление поля в список запроса и скрещение других свойств окна запроса осуществляется практически также, как при работе над созданием запроса, который основан на одной таблице.

Отчет для реляционной базы данных целостнообразно создается на запросе, включающем несколько таблиц. В общем случае можно воспользоваться мастером отчетов для данного типа запроса, так и для любого другого запроса.

Задания:

1.1 Запустить Access, создать новую базу данных «Продажи», сохранить ее на диске D:\ в папке, названной вами группой.

1.2 Создать таблицы «Клиенты», «Заказы», «Представители», «Товары», введенные в базу данных «Продажи», схема данных которой приведена на рисунке 1

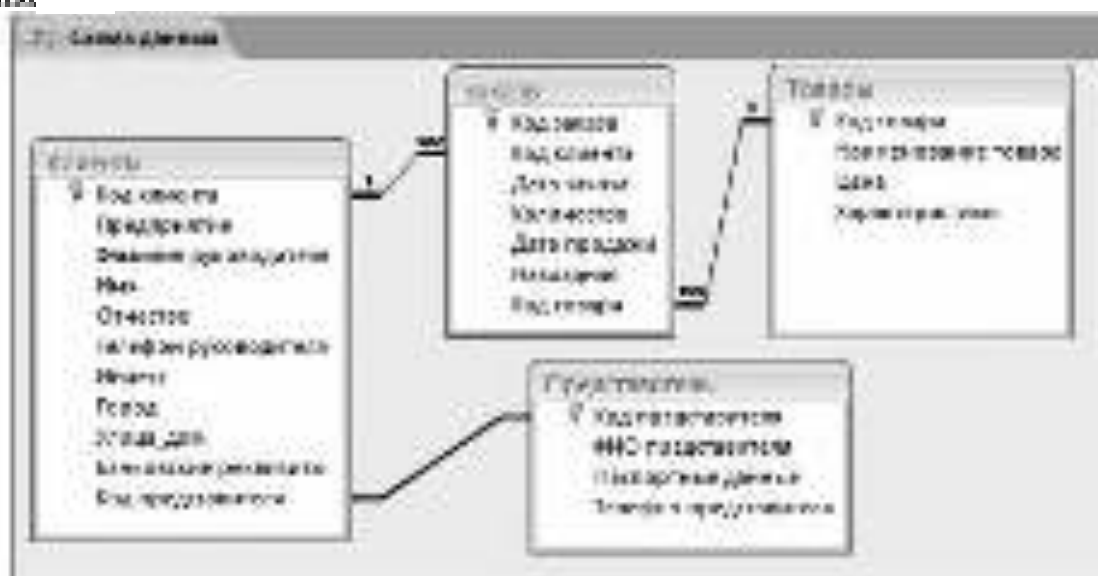


Рисунок 1 – Схема данных БД «Продажи»

3. Произвести ввод данных в данные таблицы (по 20-30 записей).
- 4 Установить связи между таблицами в окне диалогов «Связи данных», определить условия целостности данных.
- 5 Сформулировать многотабличные запросы к связанным таблицам, задавая различные условия отбора; сохранить их.
- 6 Сформулировать на основе запросов, выполняемых в пункте 1.5.
- 7 Сформулировать пользовательскую форму с подчиненной формой.
- 8 Выполнить нормализацию базы данных «Отношения между работодателями и подчиненными», структура которой представлена в таблице 1

Таблица 1 – Отношения между работодателями и подчиненными

Наименование	Тип
1 Код сотрудника	Числовой
2 Имя	Текстовый
3 Отчество	Текстовый
4 Фамилия	Текстовый
5 Дата рождения	Дата/время
6 Адрес домашний	Текстовый
7 Телефон домашний	Текстовый
8 Должность	Текстовый
9 Номер счета	Текстовый
10 Дата счѐта	Дата/время
11 Отработанный час	Числовой
12 Пенсия	Числовой
13 Состояние работы	Поле Match
14 Код работодателя	Числовой
15 Адрес предприятия	Текстовый
16 Наименование предприятия	Текстовый
17 Руководитель	Текстовый
18 Телефон предприятия	Текстовый

В отчете следует предоставить:

- I)
 - структуру данных БД «Продажи» в третьей нормальной форме с указанием типа связей между таблицами;
 - вид бланков запроса на языке QBE при различных заданных условиях отбора в многотабличных запросах, выполняемых в п. 5;
 - вид отчета, полученного в п.6;
 - макет пользовательской формы с подчиненной формой, которая создавалась при выполнении п. 7;
 - структура БД «Отношения между работодателями и подчиненными» после проведения нормализации.
- II) ответы на тестовые задания и контрольные вопросы:

Тестовые задания и контрольные вопросы:

1. Строки таблицы соответствуют:

- а) объекту;
- б) атрибуту;
- в) характеристике;
- г) признаку.

2. Постройте в соответствие схемы взаимосвязей таблиц и виды связей:

1)



2)



3)



1) М : М

2) 1 : 1

3) 1 : М

а) 1 - 1, 2 - 2, 3 - 3;

б) 1 - 2, 2 - 3, 3 - 1;

в) 1 - 3, 2 - 2, 3 - 1;

г) 1 - 3, 2 - 3, 3 - 2;

д) 1 - 3, 2 - 1, 3 - 2.

3. Связи между таблицами реляционной базы данных позволяют:

- а) избежать дублирования информации;
- б) определить местоположение пустой таблицы;
- в) производить сортировку таблицы;
- г) рассчитывать данные таблицы;
- д) удалять таблицы.

4. Какие связи не допускают реляционные многотабличные БД:

- а) многие — к одному;
- б) один — ко многим;
- в) один — к одному;

г) минимизация избытка;

5 Между двумя реляционными таблицами могут быть сформированы связи, если они имеют:

- а) одинаковые поля;
- б) одинаковые количества столбцов;
- в) одинаковые количества строк;
- г) общие первичные данные;
- д) одинаковые записи.

6 Основная цель проектирования - разработать эффективную структуру данных, что подразумевает:

- а) обеспечить быстрый доступ к данным в таблице;
- б) производить анализ средств автоматизации;
- в) осуществлять ограниченный доступ к информации;
- г) использовать интеллектуальную память.

7 Нормализация баз данных - это

- а) создание эффективной структуры данных;
- б) арифметическая операция над данными;
- в) обеспечение секретности данных;
- г) выполнение запросов к базе данных;
- д) обеспечение взаимосвязности данных.

8 Поставьте в соответствие нормальную форму и требование, предъявляемое к ней:

- 1) первая нормальная форма
- 2) вторая нормальная форма
- 3) третья нормальная форма

1) каждый атрибут таблицы, не являющийся ключевым, не транзитивно зависит от каждого возможного ключа этого отношения

- 2) поля должны быть неделимыми
- 3) поля должны зависеть от первичного ключа

- а) 1 - 1, 2 - 2, 3 - 3;
- б) 1 - 2, 2 - 3, 3 - 1;
- в) 1 - 3, 2 - 2, 3 - 1;
- г) 1 - 3, 2 - 3, 3 - 2;
- д) 1 - 3, 2 - 1, 3 - 2.

9 К требованию к третьей нормальной форме таблицы относятся:

- а) поля должны быть неделимыми;
- б) не должно быть повторяющихся значений;
- в) каждый атрибут таблицы, не являющийся ключевым, не транзитивно зависит от каждого возможного ключа этого отношения;
- г) таблица должна содержать, как минимум, три ключевых поля.

10 Зависимость между полями А и С, при которой поле С функционально зависит от поля В, а поле В функционально зависит от поля А; при этом не существует функциональной зависимости поля А от поля В, называется

- а) полной функциональной зависимостью;
 - б) зависимостью соединения;
 - в) транзитивной зависимостью;
 - г) множественной функциональностью.
- 11 Из чего состоит реляционная база данных?
- 12 Какие связи могут существовать между отдельными таблицами в базе данных?
- 13 Что такое «первичный ключ»?
- 14 Что означает отношение «один-к-одному»? Приведите пример.
- 15 Что означает отношение «многие-к-одному»? Приведите пример.
- 16 Что означает отношение «многие-ко-многим»? Приведите пример.
- 17 Каковы цели разработки эффективной структуры данных?
- 18 Что называется «нормализацией»?
- 19 Какие требования предъявляются к первой нормальной форме базы данных?
- 20 Какие требования предъявляются ко второй нормальной форме?
- 21 Какие требования предъявляются к третьей нормальной форме?
- 22 Что такое «транзитивная зависимость»?
- 23 Как в Access устанавливаются связи между таблицами?
- 24 Что можно увидеть в диалоговом окне «Связи данных»?
- 25 Какие действия можно выполнить в окне диалогов «Связи»?
- 26 Как можно изменить существующую связь?
- 27 Что означает «установка целостности данных»?
- 28 Какие ограничения действуют при определении условий целостности данных?
- 29 Для чего предназначены каскадные операции?
- 30 Как осуществляется импорт/экспорт данных?
- 31 Как формируются отчеты в реляционных базах данных?
- 32 Что необходимо сделать при работе мастера формы для создания подчиненных форм?
- 33 Для чего создается пользовательская форма с подчиненными формами?
- 34 Что содержит пользовательская форма с подчиненной формой?
- 35 Чем отличается пользовательская форма с подчиненными формами от обычной формы?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 14

Тема: Создание запросов в Access с помощью SQL 2007

Цель работы: используя синтаксис структурированного языка запросов SQL сформировать практические навыки по созданию запросов в СУБД MS Office Access 2007.

Теоретический материал:

Рост количества данных, необходимость их хранить и обрабатывать привели к тому, что возникла потребность в создании стандартного языка для данных, который мог бы функционировать в большом количестве разных аппаратно-технических систем. Такой стандартный язык повышает эффективность manipulation данных независимо от того, работает ли он на персональном компьютере, основной рабочей станции, или на универсальной ЭВМ.

SQL (Structured Query Language) – это сокращенное название структурированного языка запросов, предназначенное для работы со структурой и данными в различных базах данных (БД). Независимость от специфики компьютерных технологий, а также поддержка SQL лидерами промышленности в области технологии реляционных баз данных сделали его основным стандартом языка баз данных.

Язык SQL является основой многих СУБД, так как он позволяет задавать конкретное структурирование и запись данных на диске, а также из физического хранения данных в диске и позволяет принимать SQL-запросы от других компонентов СУБД и пользовательских приложений. SQL является основным инструментом, который обеспечивает пользователям, программистам и администраторам системы доступ к информации, хранящейся в реляционных БД.

Основные достоинства языка SQL заключаются в следующем:

а) стандартность языка SQL, – это использование в программах стандартизированной международными организациями;

б) независимость от конкретных СУБД – как распространённых СУБД используют SQL, так как реляционную БД и программы, которые с ней работают, можно перенести с одной СУБД на другую с минимальными доработками;

в) возможность переноса с одной вычислительной системы на другую;

г) реляционная основа языка;

д) возможность создания интерактивных запросов – SQL обеспечивает пользователям немедленный доступ к данным, при этом в интерактивном режиме можно получить результат запроса за очень короткое время без написания сложной программы;

е) возможность программного доступа к БД;

ж) обеспечение разнородного представления данных – с помощью SQL можно предусмотреть такую структуру данных, что пользователи будут видеть различные представления данных;

з) возможность динамического изменения и расширения структуры БД; SQL обеспечивает гибкость с точки зрения приспособленности БД к изменяю-

понижа требования предметной области, не прерывая при этом работу приложения.

Язык SQL предназначен для выполнения операций над таблицами (создание, удаление, изменение структуры) и над данными таблиц (выборка, изменение, добавление и удаление, а также некоторых сопутствующих операций. SQL является декларативным языком и не содержит операторов управления, организации программы, ввода-вывода.

SQL автономно не используется, обычно он погружен в среду встроенного языка программирования СУБД (например, FoxPro СУБД Visual FoxPro, ObjectPASCAL, SYBID, Paradox, Visual Basic for Applications SYBID Access).

В современных СУБД с интерактивным интерфейсом можно создавать запросы, используя другие средства, например, QBE (язык запросов по образцу, имитирующий манипулировать запросом к наглядной форме). Однако применение SQL зачастую позволяет повысить эффективность обработки данных в базе. Например, при подготовке запросов в среде Access можно перейти на язык конструктора запросов в одно с эквивалентным оператором SQL. Подготовку нового запроса путем редактирования уже имеющегося в ряде случаев проще выполнить путем изменения оператора SQL. В различных СУБД состав операторов SQL может несколько отличаться.

Запросом SQL называют запрос, создаваемый с помощью инструкции SQL. Примерами запросов SQL являются запросы на объединение, запросы к серверу, управляющие и подчиненные запросы.

Запрос на объединение – это такой запрос, в котором объединяются поля (столбцы) одной или нескольких таблиц или запросов в одно поле или столбец в результирующем наборе записей. Например, в пяти целях предназначены установленное стандартное оборудование. Состав запрос на объединение, можно объединить список приборов, входящих в результирующем наборе записей, а затем редактировать запрос на создание таблицы, основанной на запросе на объединение.

Запрос к серверу – выполняет передачу команд SQL-серверу. Запросы к серверу позволяют непосредственно работать с таблицами на сервере вместе их присоединение. Результатом выполнения запроса к серверу может быть загрузка записей или изменение данных.

Управляющий запрос – создает или изменяет объекты базы данных, такие как Access или SQL Server.

Подчиненный запрос – состоит из инструкции SQL SELECT, находящейся внутри другого запроса на выборку или запроса на изменение. Эти инструкции вводятся в строку «Поле» языка запроса для определения нового поля или в строку «Условия отбора» для определения условий отбора записей. Подчиненные запросы используются для выполнения следующих действий:

- проверка в подчиненном запросе существования некоторых результатов с помощью зарезервированных слов EXISTS или NOT EXISTS;

- поиск в главном запросе любых значений, которые равны, больше или меньше значений, возвращаемых в подчиненном запросе (с помощью зарезервированных слов ANY, IN или ALL);

- создание подчиненных запросов внутри подчиненных запросов (или-жечных подчиненных запросов).

Язык SQL в Access может применяться при разработке зарплатных форм, отчетов, а также при создании макросов и программ на Visual Basic for Applications (VBA). В Access для создания запросов используется также язык QBE. Между языками QBE и SQL, имеется тесная связь. Запросы, таблицы (базы), формы на языке QBE, заполняемые пользователем, перед непосредственным исполнением преобразуются в выражения (или выражения) SQL. То есть язык SQL является внутренним стандартом по выполнению запросов. Такой механизм имеет преимущество, поскольку позволяет внутри системы Access унифицировать подготовку запросов и выполнение на локальном и на удаленном компьютерах.

В Access запрос может находиться в одном из трех режимов (состояний): конструктора, SQL и таблицы. Режим конструктора применяется для разработки нового запроса с чистого листа (без использования мастеров или других средств) или для изменения листа существующего запроса. Режим SQL применяется для ввода или просмотра инструкций SQL. Режим таблицы применяется для работы с результатами выполнения запроса.

В режиме табличного запроса переходят при выборе нужного запроса в области переходов, для удобства работы предварительно сгруппировать все объекты Access по типам. Перейти в режим конструктора или режим SQL можно, выбрав соответствующий режим на раскрывающемся списке кнопки Вид в панели Главная группы «Представление».

Операторы языка SQL можно условно разделить на две подгруппы:

- язык определения данных, к нему относятся операторы CREATE TABLE (оператор создания таблицы), ALTER TABLE (оператор изменения структуры таблицы), DROP TABLE (оператор удаления таблицы), CREATE VIEW (оператор создания представления) и другие;

- язык манипулирования данными, основным оператором которого является оператор SELECT (оператор выборки записей).

В таблице 1 приведены составленные на основании при анализе статистики SO_2 .

Таблица 1 – Описание синтаксиса SQL

Составление ИДЧ	Толкование
ИЗМЕНЧИВОСТИ	Применяются функции выбора ключевых и стандартизованных слов, которые должны быть найдены в соответствии с требованиями
Анализ	Слова, которые являются синонимами, антонимами, и другими словами, которые являются синонимами
Условные свойства	Применяются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами. Также используются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами.
Условные свойства	Применяются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами. Также используются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами.
Условные свойства	Применяются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами. Также используются функции выбора ключевых слов, которые являются синонимами, антонимами, и другими словами, которые являются синонимами.

Оператор SELECT является ядром языка SQL. Он используется для отбора строк и столбцов из таблиц базы данных. При выполнении данного оператора Microsoft Jet выводит указанную таблицу или таблицы, вытаскивает заданные столбцы, выдает строки, соответствующие условиям отбора, и сортирует или группировает результирующие строки в указанном порядке. Инструкция SELECT не изменяет данные в базе данных. Она содержит пять основных предложений.

В общем случае синтаксис можно представить в следующем виде:

```
SELECT <выражения>  
FROM <таблицы>  
[WHERE <предложение>]  
[GROUP BY <предложение>]  
[HAVING <предложение>]  
[ORDER BY <предложение>]
```

Обязательными являются предложения SELECT и FROM. Все дополнительные предложения SELECT, если они используются, должны быть указаны в том порядке, в котором они перечислены в формате команды.

Предложение SELECT определяет состав результирующего набора данных, эти поля могут принадлежать разным таблицам. Оно имеет следующий формат:

```
SELECT [DISTINCT|DISTINCTROW|ALL TOP n [PERCENT]]  
[ * | <имя столбца> ]
```

Предикат DISTINCTROW указывает отбирать строки из исходных данных, выведенные в список таблиц, с различными значениями заданных значений в таблице, из которых поступает список столбцов. Если указан предикат ALL, то выводится все строки, в том числе повторяющиеся. Так как по умолчанию принимается значение ALL, то оно в явном виде указывать не требуется. Предикат DISTINCT требует, чтобы запрос возвращал только строки, отличающиеся от всех остальных.

Чтобы результирующий набор содержал только первые n или первые n процентов записей, необходимо использовать <TOP n> или <TOP n PERCENT>. Параметр должен быть целым числом, не превышающим 100, если используется ключевое слово PERCENT. Для каждой таблицы и запроса при желании можно определить альтернативное имя (псевдоним). Псевдоним используется при задании имен столбцов в списке полей, предложениях WHERE и в полученных предложениях вместо полного имени таблицы, имени запроса или для ссылки на инструкцию выбора.

Знач * указывается, если в ответ выводится все столбцы в том порядке, в котором они были заданы при описании таблицы.

В списке выражений перечисляются имена/выражения столбцов, которые будут выведены в ответ. В выражение можно включить имя столбца только той таблицы или запроса, которые указаны в предложениях FROM. Если имя таблицы, встречаемая в нескольких таблицах или запросах данной инструкции, то необходимо полностью идентифицировать, указав также имя таблицы, запроса или псевдонима. Может быть использован следующий синтаксис для <имени выражения> *:

```
<выражение> [AS имя - <идентификатор - столбец>] или - <таблица> .4 |  
или - <запрос> .5 [псевдоним].6
```

Выражение определяет некоторое значение в предикате или в списке полей инструкции SELECT. Это истинное выражение может быть заменено так-

вообще итоговая функция. В Access применяют следующие итоговые функции:

- AVG (вычисляет среднее арифметическое значение выражения или заданного поля);
- COUNT (возвращает значение, равное числу строк в результирующей таблице);
- MAX (вычисляет максимальное значение выражения или заданного поля);
- MIN (возвращает минимальное значение выражения или заданного поля);
- STDEV, STDEVP (вычисляет корень квадратный из дисперсии значений выражений или заданного поля);
- SUM (возвращает сумму значений выражения или заданного поля);
- VAR (вычисляет дисперсию значений выражения или заданного поля).

В выражении могут быть также различные формулы, по которым вычисляются значения поля в запросе. В арифметическом выражении допустимо использовать поля, содержащие числовые значения. Перед остальными выражениями указывается функция COE. Круглые скобки используются для уточнения порядка вычислений.

Предложение FROM задает таблицу или запросы, служащие источником данных для выполнения запроса. Они имеют следующий вид:

FROM (имя – таблица [[AS] <таблица>] | имя – запрос | имя – выборку
[[AS] <таблица>] | <таблица> - <объединение>);

где <таблица> - <объединение>:

(имя – таблица [[AS] <таблица>] |
имя – запрос | имя – выборку [[AS] <таблица>] |
<таблица> - <объединение>) (INNER | LEFT | RIGHT) JOIN
(имя – таблица [[AS] <таблица>] |
имя – запрос | имя – выборку [[AS] <таблица>] |
<таблица> - <объединение>) ON <условие объединения>)

В предложении FROM может быть указано только <имя – таблица>, из которой будет выбран запрос.

Обширные возможности SQL во многом основаны на способности этого языка объединять информацию из нескольких таблиц или запросов и представлять результат в виде единого логического набора записей. В большинстве случаев Access позволяет объединять набор записей запроса со объединением, как если бы это были отдельная базовая таблица.

Для задания типа объединения таблиц в логический набор записей, из которого будет выбираться необходимая информация, используется в предложении FROM операция JOIN. Можно включить в логический набор записей только соответствующие строки обеих таблиц (так называемое внутреннее объединение – inner join) или включить все строки одной из двух заданных таблиц, даже если соответствующие строки не найдены во второй таблице (левое или правое объединение – outer join). Можно использовать различные операции JOIN, например, объединить с результатом объединения двух таблиц третью.

Используя операцию INNER JOIN, можно получить все строки из обеих логических таблиц, удовлетворяющих условию объединения (логическая таблица – это любая таблица, запрос или таблица объединения). Операция LEFT JOIN возвращает все строки из первой логической таблицы, объединенные с теми строками из второй, для которых выполняется условие объединения. Если во второй логической таблице нет таких строк, Access возвращает значение Null в столбцах второй таблицы. Аналогично, операция RIGHT JOIN возвращает все строки из второй логической таблицы, объединенные с теми строками из первой таблицы, для которых выполняется условие объединения.

Если в условии объединения используется только оператор равно (=), результат называется объединением по равенству. В бланке запроса можно задать лишь одно. Запрос на объединение таблиц по неравенству (<, >, <=, >= или <>) можно создать только в режиме SQL.

Предложение WHERE задает условие отбора в инструкции или предложении SQL. Инструкции DELETE, SELECT и UPDATE и подчиненный запрос, содержащие предложение WHERE, воздействует только на те строки, которые удовлетворяют условию отбора.

Синтаксис предложения:

WHERE <условие отбора>

Access применяет условие отбора в каждой строке логической таблицы, полученной в результате выполнения предшествующих предложений, и открывает те, для которых условие отбора не принимает значения TRUE.

Если в условии отбора используется подчиненный запрос внутри предложения (подчиненный запрос в этом случае часто называют *вложенным запросом*), то Access сначала выполнит подчиненный запрос и только потом определит значение предложения. Если подчиненный запрос согласован по таблице или запрос, используемый во вложенном предложении FROM (в таком случае его обычно называют *связанным вложенным запросом*), то Access выполняет подчиненный запрос для каждой обрабатываемой строки внешней таблицы. Если в подчиненном запросе нет ссылок на внешнюю таблицу, то Access выполняет его только один раз. Связанный подчиненный запрос может быть записан как запрос на объединение, который обычно более эффективен.

В предложении WHERE можно установить несколько условий, которым должны удовлетворять поля запроса, в этом случае используются логические связи. Логические операторы в условии отбора выполняются в следующем порядке: NOT, AND, OR, XOR (исключение OR), EQV (равенство) и IMP (импликация). Для изменения порядка вычисления логических выражений можно использовать круглые скобки. В выражениях в условии отбора могут применяться итоговые функции AVG, COUNT, MAX, MIN и другие.

В выражении условия отбора может использоваться предикат BETWEEN, который сравнивает значение с заданным диапазоном. Он имеет синтаксис:

<выражение> <[NOT] BETWEEN <выражение> AND <выражение>

Типы данных выражений должны быть идентичны. Сравнение буквенно-цифровых литералов (строк) в Access производится без учета регистра.

Предложение GROUP BY в команде SELECT задает столбцы, используемые для формирования групп из выбранных строк. Строки каждой группы содержат одно и то же значение заданного столбца (столбцов). В Access пред-

пожение используется для задания итерационных запросов. Оно имеет следующий синтаксис:

GROUP BY (имя – столбец, ...)?

Имя столбца в предложении **GROUP BY** может быть именем произвольного столбца из любой таблицы, упомянутой в предложении **FROM**, даже если этот столбец не содержится в списке полей инструкции **SELECT**. Если предложение **GROUP BY** расположено после предложения **WHERE**, Access создает группы из строк, выбранных после применения предложения **WHERE**. При включении предложения **GROUP BY** в инструкцию **SELECT** список полей должен состоять из итерационных функций SQL (**AVG**, **COUNT**, **MAX**, **MIN**, **STDEV**, **STDEVP**, **SUM**, **VAR**, **VARP**) или из имен столбцов, указанных в предложении **GROUP BY**.

Предложение **ORDER BY** задает порядок расположения строк, возвращаемых инструкцией **SELECT** или **INSERT**. Оно имеет синтаксис:

ORDER BY (имя – столбец) [номер столбца [**ASC** | **DESC**]], ...

Для задания столбца, по значениям которого упорядочиваются возвращаемые строки, можно использовать имя столбца или его идентификационный порядковый номер в наборе данных запроса (первый возвращаемый столбец имеет номер 1). В предложении **ORDER BY** можно указать несколько столбцов. Список сортируется сначала по значениям столбца, имеющего указанный первый. Строки с равными значениями в этом столбце упорядочиваются по значениям столбца, имя которого находится на второй позиции в списке предложения **ORDER BY**. Для каждого столбца можно задать порядок сортировки по возрастанию (**ASC**) или по убыванию (**DESC**). Если порядок не указан, по умолчанию применяется сортировка по возрастанию. Использование предложения **ORDER BY** в инструкции **SELECT** – единственный способ задания последовательности, в которой располагаются возвращаемые значения.

Задания:

1. В папке под своей фамилией создать папку «ПР 14».
2. Создать БД «Поступление товаров», в которой последовательно создать следующие таблицы:
 - Поставщики;
 - Типы товаров;
 - Товар;
 - Склад.
3. В диалоговом окне «Схема данных» установить связи между таблицами.

Таблица 1 - Поставщики

Код поставщика	Полное название	Адрес	Телефон	WWW-адрес
1	Компьютерный салон «Центр»	ул. Пролетарская, 42	734311	http://www.komputer.ru
2	Компьютерный магазин «Bayer»	ул. Дзержинского, 20	363582	http://www.bayer.ru
3	ООО «Компани» «Микроэлектроника»	ул. Гага, 5	780757	http://www.Micro.ru
4	Компьютерный салон «Гладио»	ул. Воскресенского, 10	777665	http://www.Gladio.net
5	ТЦ «Дельта»	ул. Воскресенского, 27	770268	http://www.juno-computer.com.ru
6	ООО «Каскады» - Сервис»	пр-т Парковый, 46	734364	http://www.Cascadya.ru

Таблица 2 - Типы товаров

Код типа	Полное название	Продукты
10	Монитор	Samsung
15	Монитор	LCD Acer
20	Монитор	LCD LG
25	Принтер	Samsung
30	Принтер	HP
35	Процессор	Intel Celeron
40	Процессор	AMD

Таблица 3 - Товары

Код товара	Тип товара	Технические характеристики	Поставщик	Цена
100	10	17" T23N AKS «Silver» (LCD, 1280x1024)	4	1880
102	20	17" L1734S-11N Dutton «Black» (LCD, 1280x1024)	1	7607
104	20	LaserJet P1505 «CE410A» A4 140/мин 2945 USB2.0	3	4199
106	40	CPU AMD ATHLON-64 X2 4200+ (AD64200) 1902 1000MHz Socket AM2	4	1739
110	35	CPU Intel Celeron 430 1.8 ГГц/ 512K/ 800MHz 775-LGA	5	1154
112	15	17" AL1716Pa (LCD, 1280x1024)	5	1900
114	25	ML-1640 (A4, 80b, лазерный, 18 c./мин, 1200dpi, USB2.0)	6	3570
116	35	CPU Intel Celeron D 336 2.8 ГГц/ 256K/ 533MHz 775-LGA	1	1910
118	10	19" V920 50V «Black» (LCD, 1280x1024, 115V)	1	7670
120	15	19" MONITOR Acer V1930m «Black» (LCD, 1280x1024)	1	4080
122	25	SCX-4200 (18 c./мин, 80b, лазерный, A4, копир, сканер, USB2.0)	3	6147
124	20	hp LaserJet P1505 «CE410A» A4 330/мин 2945 USB2.0	4	6515

126	40	CPU AMD Phenom X3 8450 BOX (HD8450) 1.5 (2MB) 3800MHz Socket AM2+	2	7430
128	35	CPU Intel Pentium 4 631 3.0 GHz 2MB 800MHz 775 LGA	1	2144
130	35	CPU Intel Core 2 Duo E6550 2.33 GHz 4MB 1333MHz 775 LGA	2	5187
132	30	Lenovo P2014 <CB455A> A4, 23 c/часа 32Mb USB2,DLPT	4	9675
136	15	20" V700Wh <Black> (LED, Wide, 1680x1050)	4	5380
138	20	19" L1942S-BE Flatron <Black> (LCD, 1280x1024)	4	6480
140	10	19" 431NW MCS <Silver> (LCD, Wide, 1440x900)	1	4770

Таблица 4 – Склад

Код товара	Дата поступления	Количество
103	15.01.09	5
130	17.01.09	10
108	17.01.09	12
106	18.01.09	8
104	20.01.09	5
110	25.01.09	3
124	26.01.09	4
112	27.01.09	10
140	12.02.09	15
128	15.02.09	10

5 Создать SQL – запросы, которые определит:

- пять самых дешевых товаров;
- товары с ценой ниже средней;
- число товаров, поступивших на ТЦ «Джам»;
- перечень наименее дорогих товаров на складе и их стоимость.

Формулировку созданного SQL – запросов записать в тетрадь.

В отчете следует представить:

- логическую схему БД «Поступление товаров»;
- перечень команд, использованных для выполнения практической работы, с указанием их назначения;
- формулировку SQL-запросов, созданных по заданию 5;
- ответы на тестовые задания и контрольные вопросы.

Тестовые задания и контрольные вопросы:

1 Язык SQL – это:

- структурированный язык отчетов;
- система управления базой данных;
- структурированный язык запросов;
- язык иерархический базы данных;
- структурированный язык форм.

2. Оператор **SELECT** относится к:

- а) языку определения данных;
- б) языку манипулирования данными;
- в) языку описания данных;
- г) языку ввода данных;
- д) языку отчетов.

3. К достоинствам языка **SQL** не относится:

- а) стандартизация;
- б) разграничение доступа к языку;
- в) минимальный **TPC/TJCL**;
- г) возможность программного доступа;
- д) возможность создания интерактивных запросов.

4. В угловые скобки, используемые при описании синтаксиса языка **SQL**, включаются:

- а) слова, соответствующие переменным, которые заданы пользователем;
- б) обязательные элементы синтаксиса;
- в) элементы, из которых выбирается один;
- г) элементы, которые повторяются;
- д) необязательные элементы.

5. В квадратные скобки, используемые при описании синтаксиса языка **SQL**, включаются:

- а) слова, соответствующие переменным, которые заданы пользователем;
- б) обязательные элементы синтаксиса;
- в) элементы, из которых выбирается один;
- г) элементы, которые повторяются;
- д) необязательные элементы.

6. В фигурные скобки, используемые при описании синтаксиса языка **SQL**, включаются:

- а) слова, соответствующие переменным, которые заданы пользователем;
- б) обязательные элементы синтаксиса;
- в) элементы, из которых выбирается один;
- г) элементы, которые повторяются;
- д) необязательные элементы.

7. Задание значения **DESC** в предложении **ORDER BY** оператора **SELECT** означает:

- а) группировку по столбцам;
- б) сортировку по возрастанию;
- в) выборку данных;
- г) сортировку по убыванию;
- д) группировку по строкам.

8. Задание значения **ASC** в предложении **ORDER BY** оператора **SELECT** означает:

- а) группировку по столбцам;
- б) сортировку по возрастанию;
- в) выборку данных;
- г) сортировку по убыванию;
- д) группировку по строкам.

9. Функция **AVG** при построении выражения в операторе **SELECT** возвращает:

- а) значение, равное числу строк в результирующей таблице;
- б) среднее арифметическое значение выражения или заданного поля;
- в) максимальное значение выражения или заданного поля;
- г) сумму значений выражения или заданного поля;
- д) дисперсию значений выражения или заданного поля.

10. Нисовые функции COUNT при построении выражения в запросе SELECT возвращает:

- а) значение, равное числу строк в результирующей таблице;
- б) среднее арифметическое значение выражения или заданного поля;
- в) максимальное значение выражения или заданного поля;
- г) сумму значений выражения или заданного поля;
- д) дисперсию значений выражения или заданного поля.

11. Что такое SQL?

12. В чем заключаются основные достоинства языка SQL?

13. Для чего предназначен SQL?

14. Что означает запросы SQL?

15. Приведите примеры запросов SQL, выполняемых в MS Access.

16. Как перейти в Режим SQL?

17. На какие два подтипа можно условно разделить операторы языка SQL?

18. Для чего предназначен оператор SELECT?

19. Что включает синтаксис оператора SELECT?

20. Какие нисовые функции применяются в запросах?

21. Что задает предложение FROM?

22. Какие операторы в предложении FROM используются для задания типа объединения таблиц и логической выборки записей?

23. Что задает предложение WHERE?

24. Какие логические операторы могут использоваться в предложении WHERE?

25. Что используется для изменения порядка выполнения логических выражений в предложении WHERE?

26. Что задает предложение GROUP BY?

27. Что задает предложение ORDER BY?

Задание на дом:

Выполнить отчет по практической работе.

Практическая работа № 15

Тема: Разработка приложений в MS Office Access 2007

Цель работы: создание пользовательского приложения по работе с базой данных в MS Office Access 2007.

Теоретический материал:

Приложение представляет собой программу или комплекс программ, обеспечивающих автоматизацию обработки информации для прикладной задачи.

Приложение базы данных – программа или комплекс программ, использующих базу данных и обеспечивающих автоматизацию обработки информации из некоей предметной области. Приложение базы данных в Access содержит набор объектов, среди которых могут быть таблицы, запросы, формы, отчеты, макросы и модули кода, которые используются вместе и упрощают использование базы данных.

Access относится к средствам быстрой разработки приложений (ВБР – Rapid Application Development). Можно выделить следующие отличительные черты таких средств разработки:

- наличие объектно-ориентированного языка программирования, позволяющего эффективно использовать модульный принцип построения программ;
- использование визуальных средств разработки, предоставляющих возможность избежать написания программного кода рисованием пользовательского интерфейса и заданием необходимой функциональности анимационными средствами;
- поддержка стандартных протоколов обмена данными между приложениями, позволяющая разрабатывать многоуровневые приложения, не зависящие от источника данных.

Паряду с созданием обычных ACCDB-файлов приложений в Access 2007 (в более ранних версиях Access – MDB-файлов) имеется возможность создавать ACCDE-файлы приложений (в более ранних версиях Access – MDE-файлов), в которых хранятся базы данных, предназначенные только для выполнения. Отличительной чертой этих файлов является то, что в соответствии с предложением нельзя использовать конструкторы для модификации приложения, а также код модуля является закомпилированным. Это позволяет защитить приложение от исправления и использования, а также уменьшить размер файла базы данных.

Чтобы создать копию приложения с базой данных, предназначенную только для выполнения, нужно выполнить команду Создать ACCDE, находящуюся на Ленте на вкладке Работа с базой данных в группе «Работа с базой данных». Появится аналогичное окно «Сохранить как», в котором вводится имя создаваемого файла и папка его размещения, затем необходимо нажать кнопку Сохранить. Access скомпилирует исходную базу и сохранит ее в новом файле с расширением ACCDE, после чего выполнит сжатие файла.

При открытии ACCESS-файла некоторой базы на Ленте на панели Создания станут недоступными (именно свой пункт на серой) команды групп «Формы» и «Отчеты», команда Конструктор запросов в группе «Другие», а также нельзя открыть созданные формы и отчеты в режиме конструктора. Это означает, что нельзя изменить или создать формы, отчеты и модули. Вместе с тем по-прежнему останутся доступными все средства просмотра, создания и модификации таблиц, запросов и макросов.

Меню в прикладной программе – это первое, что видит пользователь, решив запустить приложение.

Основное назначение меню заключается в том, чтобы дать возможность пользователю получить легкий доступ ко всем элементам прикладной программы. При разработке меню придерживаются следующих принципов:

- заголовок должен включать максимально полную информацию о его назначении;
- структура меню должна соответствовать частоте выполняемых действий, значимости полнотекстовости их наименования или, в крайнем случае, хотя бы по алфавитному порядку;
- функционально связанные группы команд обозначают с помощью разделителей.

В Access создается меню меню к виде кнопочной формы. Построение всех кнопочных форм, необходимых для управления сложным приложением, может потребовать довольно трудоемкого процесса. Специальная надстройка Access – диспетчер кнопочных форм – помогает выполнить эту работу. Эта надстройка применяет довольно сложную технику для управления всеми кнопочными формами с помощью одной формы и использует специальную таблицу – драйвер с именем «Элементы кнопочной формы», что позволяет определить любое число кнопочных форм и создать по какому-либо кнопке в каждой из них.

Чтобы запустить надстройку, необходимо выбрать на Ленте на панели Работа с базами данных в группе «Работа с базами данных» команду Диспетчер кнопочных форм (рисунк 1). Диспетчер кнопочных форм сначала проверит, есть ли в базе данных (БД) кнопочная форма и таблица Элементы кнопочной формы. В случае их отсутствия выведется окно сообщения «Не удалось найти кнопочную форму в этой базе данных. Создать кнопочную форму?». Следует нажать кнопку Да. Построить скелет главной кнопочной формы и таблицу Элементы кнопочной формы, Диспетчер кнопочных форм выведет на экран свое основное окно «Диспетчер кнопочных форм». В этом окне можно создать дополнительные кнопочные формы, изменить существующие формы, удалить их, а также закрыть форму. Чтобы построить дополнительную кнопочную форму (в диспетчере она называется страницей), необходимо щелкнуть на кнопке Создать, ввести её имя в следующем окне диалогом и щелкнуть на кнопке ОК (рисунк 2).

После создания всех необходимых дополнительных кнопочных форм необходимо выбрать одну из них в основном окне диспетчера и щелкнуть на кнопке Изменить. На экране появится окно, аналогичное тому, которое назо-

дится на рисунке 3. Здесь можно определить новый элемент кнопочной формы, определить, существует ли или изменить порядок их расположения. Раскрывающийся список Команда позволяет назначить действие для создаваемого или изменяемого элемента: переход в другой кнопочной форме, открытие формы в режиме добавления или редактирования, открытие отчета, изменение кнопочной формы, выход из приложения и другие. После выбора команды и при необходимости указания ее аргументов диспетчер помещает в кнопочную форму кнопку, после нажатия на которой будет выполняться заданная команда.

В главной кнопочной форме следует добавить кнопки, открывающие другие формы, и вывести кнопку для выхода из приложения. В каждой дополнительной кнопочной форме надо предусмотреть одну кнопку для возвращения на предыдущий уровень в иерархии кнопочных форм или для перехода в главную кнопочную форму.

После щелчка по кнопке Закреть в основном окне диспетчера кнопочных форм Ассистент переходит в текущей БД форму с именем Кнопочная форма. Там можно переименовать, изменить значение свойства «Полнота» на значение «Максимум» или свойства формы. Открыть «Окно свойств» формы можно, выбрав команду Страница свойств в группе «Сервис», доступную на Ленте в режиме конструктора формы (выкладка Конструктор).




Рисунок 1 Расположение Диспетчера кнопочных форм на Ленте



Рисунок 2 – Создание дополнительной кнопочной формы



Рисунок 3 - Создание нового элемента шнупочной формы

Поскольку, как и в программах из семейства Microsoft Office, можно зайти в режим автоматического запуска базы данных. Удобно для этих целей использовать параметры запуска и задать начальную форму приложения. Для этого необходимо нажать кнопку Microsoft Office , а затем кнопку Параметры Access. В появившемся диалоговом окне «Параметры Access», нажав на кнопку: Таблица базы данных, можно задать параметры запуска базы данных, как показано на рисунке 3.4.

В области «Параметры приложения» можно задать заголовок приложения и его значок. Можно указать, какой файл значка, если указан, что при открытии базы данных он будет выводится в появившейся форме. В противном случае надо задать полный путь к этому файлу. Поле «Форма просмотра» позволяет выбрать форму, которая будет выводится на экран при открытии базы данных. Флажок «Скрывать строку состояния» снимается, если необходимо, чтобы при запуске приложения автоматически скрывалось окно строки состояния. Настройка удобного интерфейса пользователя может осуществляться также и другими параметрами в области «Параметры приложения». «Переслать», «Параметры меню и панели инструментов», «Параметры автоматического» также «Параметры автофильтра».

Настройка внешнего вида таблицы в Access будет и шрифт по умолчанию, эффект сетки и ячеек) осуществляется по нажатию кнопки Таблица в окне диалогового «Параметры Access». Настройка параметров создания и именования объектов базы данных – по нажатию кнопки Структура объектов.



Рисунок 1.4 Установка параметров базы данных

Задания:

1. В папке под своей фамилией создать папку «ПР 15».
2. Создать БД «Датчики», в которой последовательно создать следующие таблицы, в соответствии с данными рисунков 1-3:
 - Датчики;
 - Типы датчиков;
 - Применение.

Датчики		
	Имя поля	Тип данных
1	код датчика	Смешанный
	код типа	Числовой
	название	Текстовый
	назначение	Текстовый
	код области	Числовой
	цена	Денежный
	количество	Числовой
	фото	Поле объекта OLE

Рисунок 1 - Структура таблицы «Датчики»

область применения		
	Имя поля	Тип данных
?	код области	Числовой
	область	Текстовый

Рисунок 2 - Структура таблицы «Область применения»

типы		
	Имя поля	Тип данных
?	код типа	Числовой
	тип	Текстовый

Рисунок 3 - Структура таблицы «Типы»

3. Произвести ввод данных в таблицы, сформировав в первой таблице 20 записей; во второй – 4; в третьей – 6.

Например:

Название	Тип	Назначение	Область
MiniSightPlus	Температуры	Измерение температуры	Промышленность
DewPro MMY 30	Влажности	Измерение влажности	Торговля

В практической работе поле «Фото» в таблице «Датчики» является полем объекта OLE. Создать его содержимое удобно следующим образом: для описываемого объекта сделать поле текущим, затем из контекстного меню выполнить команду «Вставить объект ...». В появившемся диалоговом окне выбрать тип объекта, а также способ его определения: создать объект с помощью соответствующей программы или вставить его контент из файла. Независимо от способа определения объекта, существуют две варианта размещения объекта в поле: запись файла (записывается с помощью кнопки **Связь**); путем размещения непосредственно объекта в базу данных или путем скопирования (флажок **Связь** включается), когда устанавливается связь между отдельным триггерным файлом объекта и записью базы данных.

4. Установить связи между таблицами в окне диалогов «Схема данных» (рисунок 4.1. Для каждой связи в диалоговом окне «Изменение связей» установить:

- флажок напротив опции **обеспечение целостности данных**, который означает, что перед тем как внести данные в подчиненную таблицу, программа будет проверять их на соответствие главной;
- флажок напротив опции **каскадное обновление связанных полей**, который означает, что изменения в главной таблице автоматически будут влиять на подчиненную;



Рисунок 4 - Схема данных для таблиц «Типы», «Датчики» и «Область применения»

– флажок напротив опции каскадное удаление связанных полей, по которой поля, удаленные в главной таблице, будут удалены и в подчиненной.

5 Создать с помощью конструктора экранную форму «Датчики», в качестве источников для которой взять таблицы «Датчики», «Типы» и «Область применения». Пример экранной формы приведем на рисунке 5

Рисунок 5 Экранная форма «Датчики»

6 Создать с помощью Конструктора следующие запросы:

- по датчикам, применяемым в промышленности;
- по датчикам, применяемым в торговле;

- по датчикам, применяемым в охране;
- по назначению (запрос с параметром).

Для запросов по различным областям применения используются все три таблицы, условие отбора записывается по полю «Область применения» из таблицы «Область применения». На экран выводится значения всех полей, кроме поля «Область применения».

Access позволяет создавать запросы, в которых не определено, какие именно значения должны использоваться при выполнении запроса. Для этого включается в запрос параметр, и при каждом выполнении запроса Access будет запрашивать конкретное условие отбора. Чтобы определить параметр, вводится в строку Условие отбора вместо конкретного значения имя или адрес, заключенные в квадратные скобки ([]). То, что заключено внутри квадратных скобок, Access рассматривает как имя параметра. Оно выводится в окне диалога при выполнении запроса. При вводе запроса по назначению к полю Конструкция необходимо сделать пометку выводу на экран из таблицы «Назначение», «Назначение», «Цена» из таблицы «Датчики». В строку Условие отбора для поля «Назначение» необходимо ввести [Введите назначение].

7 По созданным запросам в пункте 6 создать экранные формы, включая все поля запросов.

8 Создать в Мастере отчетов отчет по датчикам по основанном запросе «Датчики, применяемые к промышленностям». В отчет не включать поле «Объект», группировку осуществлять по полю «Тип». Группировку по категориям выключить из поля «Назначение». В режиме Конструктор, открыв созданный отчет, изменить название отчета, разбить отчет на отображаемые в отчете поля так, чтобы были видны все значения полей, и весь отчет помещался при просмотре на экране. Примерный вид созданного отчета показан на рисунке 6

9 При помощи Диспетчера кнопочных форм создать главную и дополнительные кнопочные формы. На главной кнопочной форме «Информационная система» (рисунк 7) при нажатии на кнопку Датчики должна открываться в режиме редактирования форма «Датчики»; на кнопку Запросы открываться дополнительные кнопочная форма «Запросы», приводящая к открытию экранных форм, созданных в пункте 6; на кнопку Отчеты – дополнительная кнопочная форма «Отчеты», приводящая к открытию отчета по датчикам, применяемым в промышленности; на кнопку Выход осуществляется выход из приложения.

10 Редактировать кнопочные формы в режиме Конструктор: изменить рисунок, изменить свойства, размер на командой Страница свойств, нажать на Ленте на вкладке Конструктор в группе «Сценарии».

11 Задать параметры запуска приложения.

12 Создать ACCDE-файл, как описано в теоретическом материале данной практической работы.

Тип	Назначение	Цена	Качество
давления	Поль Рес-1000-01	1 000 000 р.	10
	Преобразователь давления стандарт	1 500 000 р.	14
температуры	Преобразователь температуры J200-01	1 000 000 р.	16
	Анализатор-01	500 000 р.	18
уровня	Специальный датчик температуры	1 500 000 р.	17
	Уровень-01	100 000 р.	11

Рисунок 6 – Пример отчета

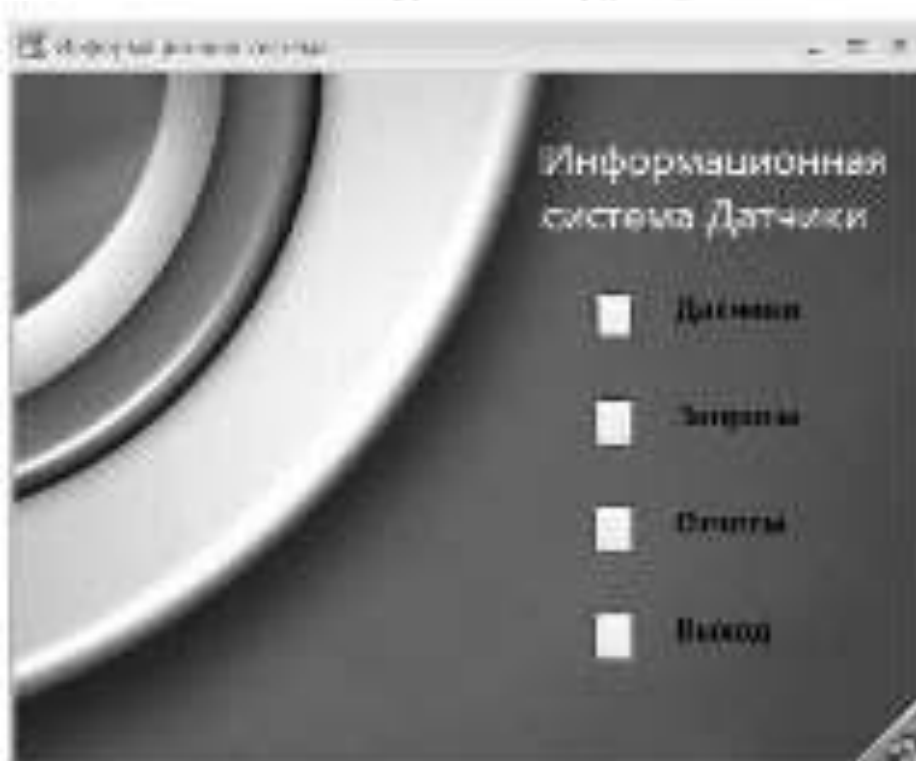


Рисунок 7 – Главная кнопочная форма

В отчете следует представить:

- перечень элементов, использованных для создания приложения, с указанием их назначения и перечнем значений определяемых свойств;

- перечень команд, использованных для выполнения практической работы, с указанием их назначения;
- размеры созданных ACCDB- и ACCDE-файлов;
- ответы на тестовые задания и контрольные вопросы.

Тестовые задания и контрольные вопросы:

1 Приложение базы данных – это:

- а) программа или комплекс программ, обеспечивающая автоматизацию обработки информации для прикладной задачи;
- б) специальным образом описанное требование, определяющее состав производимых над базой данных операций по выбору или модификации транзактных данных;
- в) объект базы данных, описанное наименование которого описанию и выводу помещено документом на основе данных базам;
- г) программа или комплекс программ, использующая базу данных и обеспечивающая автоматизацию обработки информации из некоторой предметной области;
- д) объект базы данных, в котором разработчик размещает элементы управления, служащие для ввода, отображения и хранения данных в полях.

2 Что не относится к отличительным чертам средств быстрой разработки приложений:

- а) наличие объектно-ориентированного языка программирования;
- б) поддержка стандартных протоколов обмена данными между приложениями;
- в) использование визуальных средств разработки;
- г) обеспечение автоматизации обработки информации из некоторой предметной области?

3 На какой вкладке Ленты находится команда Создать ACCDE:

- а) Создание;
- б) Работа с базами данных;
- в) Главная;
- г) Рецензия таблицы;
- д) Конструктор?

4 Особенности ACCDE-файлов заключаются:

- а) запрет на использование конструкторов форм и отчетов, баз данных, по сравнению с ACCDB-файлами, размер;
- б) запрет на использование конструкторов форм и отчетов, таблиц, по сравнению с ACCDB-файлами, размер;
- в) возможность модификации форм и отчетов, баз данных, по сравнению с ACCDB-файлами, размер;
- г) возможность модификации форм и отчетов, таблиц, по сравнению с ACCDB-файлами, размер;
- д) защита приложений от просмотра и изменения, баз данных, по сравнению с ACCDB-файлами, размер.

5 Основное назначение меню заключается в том, чтобы дать возможность пользователю получить легкий доступ:

- а) ко всем формам прикладной программы;
- б) ко всем формам и отчетам приложения;
- в) ко всем элементам прикладной программы;
- г) ко всем формам приложения.

6 На какой вкладке Ленты находится команда Двухпетчер кнопочных форм:

- а) Главная;
- б) Конструктор;
- в) Создание;
- г) Режим таблицы;
- д) Работа с базами данных?

7 Для чего используется граббер с именем «Элементы аннотационной формы»:

- а) для управления кнопочной формой открытая приложения;
- б) для вызова кнопочной формы открытое приложение;
- в) для управления всеми кнопочными формами;
- г) для аннотирования некоторой базы данных;
- д) для запуска приложения?

8 На какой вкладке Ленты находится команда Страница свойств:

- а) Конструктор;
- б) Работа с базами данных;

- в) Создание;
- г) Режим таблицы;
- д) Главная?

9 Какого типа должно быть поле, значение которого представлено картинкой, созданной приложением Windows:

- а) текстового;
- б) типа объекта OLE;
- в) гиперссылки;
- г) поле Memo?

10 При задании запроса с параметром в строку Условие отбора выводится на экран форма должна быть выполнена:

- а) в квадратные окошки;
- б) в фигурные окошки;
- в) в угловые окошки;
- г) в значимые квадратные окошки;
- д) в символы.

11 Что такое «спрашиваемые»?

12 Что такое «спрашиваемые базы данных»?

13 Что содержит приложение базы данных в Access?

14 Назовите отличительные черты средства быстрой разработки приложений?

15 Чем отличаются ACCDE- файлы, создаваемые в Access?

16 Как создать ACCDE-файл?

- 17 Для чего предназначено меню?
- 18 Какие принципы придерживаются при разработке меню?
- 19 В каком виде создается меню в лабораторной работе?
- 20 Для чего пишется драйвер с именем «Экземпляр анимации данных»?
- 21 Как создать содержимое поля объекта ОДБ?
- 22 Для чего используется команда Схема данных?
- 23 Как создается схема данных?
- 24 Что можно делать над данными и запросом?
- 25 Как создается запрос с параметром?
- 26 Для каких целей используется команда Диспетчер кнопочных форм?
- 27 Что отрисовывается в таблице «Элементы кнопочных форм»?
- 28 Что входит в параметры запуска базы данных?
- 29 Как изменять свойства формы?
- 30 Как изменить параметры приложения?

Задание на дом:

Выполнить отчет по практической работе.

Список рекомендованной литературы

Основные источники:

1. Базы данных [Электронный ресурс]: учебно-методическое пособие для выполнения лабораторных работ для студентов III курса обучающихся по направлению подготовки 231300.62 Прикладная математика/ Темирова Л.Г.— Электрон. текстовые данные.— Черкесск: Северо-Кавказская государственная гуманитарно-технологическая академия, 2014.— 57 с.— Режим доступа: <http://www.iprbookshop.ru/27177>.— ЭБС «IPRbooks», по паролю
2. Базы данных освоение работы в MS Access 2007 [Электронный ресурс]: электронное пособие/ Борзунова Т.Л., Горбунова Т.Н., Дементьева Н.Г.— Электрон. текстовые данные.— Саратов: Вузовское образование, 2014.— 148 с.— Режим доступа: <http://www.iprbookshop.ru/20700>.— ЭБС «IPRbooks», по паролю
3. Советов, Б. Я. Базы данных : учебник для СПО / Б. Я. Советов, В. В. Цехановский, В. Д. Чертовской. — 2-е изд. — М. : Издательство Юрайт, 2017. — 463 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-01653-6. — Режим доступа : <http://www.biblio-online.ru>- ЭБС «Юрайт»
4. Илюшечкин, В. М. Основы использования и проектирования баз данных : учебник для СПО / В. М. Илюшечкин. — испр. и доп. — М. : Издательство Юрайт, 2017. — 213 с. — (Серия : Профессиональное образование). — ISBN 978-5-534-01283-5. — Режим доступа: <http://www.biblio-online.ru>- ЭБС «Юрайт»
5. Лазицкас Е.А. Базы данных и системы управления базами данных [Электронный ресурс]: учебное пособие / Е.А. Лазицкас, И.Н. Загумённикова, П.Г. Гилевский. — Электрон. текстовые данные. — Минск: Республиканский институт профессионального образования (РИПО), 2016. — 268 с. — 978-985-503-558-0. — Режим доступа: <http://www.iprbookshop.ru/67612.html>

Дополнительные источники:

1. Башмакова Е.И. Создание и ведение баз данных в MS ACCESS [Электронный ресурс] : методические указания к практическим занятиям / Е.И. Башмакова. — Электрон. текстовые данные. — М. : Московский гуманитарный университет, 2014. — 46 с. — 978-5-906768-20-9. — Режим доступа: <http://www.iprbookshop.ru/39693.html>
2. Королёв В.Т. Технология ведения баз данных [Электронный ресурс] : учебное пособие / В.Т. Королёв, Е.А. Контарёв, А.М. Черных. — Электрон. текстовые данные. — М. : Российский государственный университет правосудия, 2015. — 108 с. — 978-5-93916-470-2. — Режим доступа: <http://www.iprbookshop.ru/45233.htm>

3. Самуйлов С.В. Базы данных [Электронный ресурс] : учебно-методическое пособие для выполнения лабораторной и контрольной работы / С.В. Самуйлов. — Электрон. текстовые данные. — Саратов: Вузовское образование, 2016. — 50 с. — 2227-8397. — Режим доступа: <http://www.iprbookshop.ru/47276.html>