

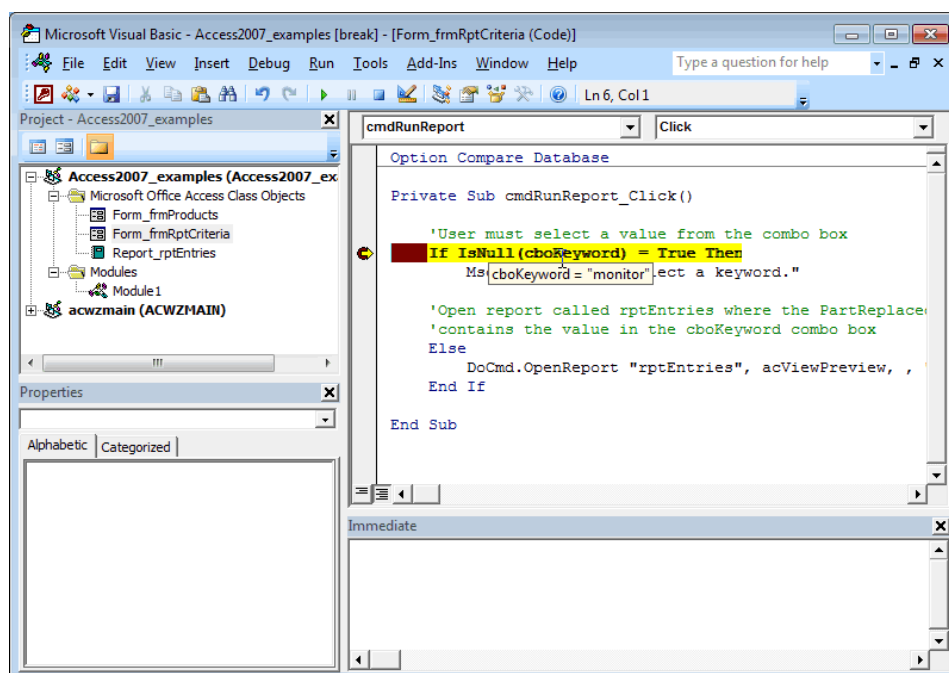
ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал
ордена Трудового Красного Знамени федерального государственного
бюджетного образовательного учреждения высшего образования
"Московский технический университет связи и информатики"



Методические указания
к лабораторным и практическим занятиям

ТЕХНОЛОГИИ БАЗ ДАННЫХ

Создание приложений БД



Ростов-на-Дону
2019

УДК 681.3.06 (076)
ББК 32.07

Чикалов А.Н. Технологии баз данных. Создание приложений БД. Методические указания к лабораторным и практическим занятиям. Ростов-на-Дону: Северо-Кавказский филиал МТУСИ, 2019.- 18 с.

В пособии изложены методические рекомендации, содержательные материалы и контрольные задания для проведения лабораторных и практических занятий по разработке приложений для баз данных. В качестве инструмента использована среда разработки MS Access и язык VBA и SQL, среда СУБД MySQL. Пособие содержит необходимые справочные материалы.

Методические указания предназначены для студентов, обучающихся по направлению подготовки 09.03.01 Информатика и вычислительная техника, профилей Вычислительные машины, комплексы, системы и сети, Программное обеспечение и интеллектуальные системы.

Пособие предназначено для использования при изучении дисциплин Технологии баз данных, а также может быть использовано преподавателями и студентами при изучении родственных дисциплин и в процессе самостоятельной работы.

Учебное пособие обсуждено и одобрено на заседании кафедры ИВТ
Протокол №1 от 26.08.2019

Рецензент Зав. кафедрой ИВТ д.т.н. профессор Соколов С.В.

СОДЕРЖАНИЕ

1. Управление базой данных с помощью VBA	4
1.1. Создание формы для импорта данных из таблицы другой БД .	6
1.2. Построение процедуры для обработки данных	9
2. Работа с базой данных на языке SQL	13
2.1. Создание и редактирование таблиц	17
2.2. Разработка запросов к БД	18

1. УПРАВЛЕНИЕ БАЗОЙ ДАННЫХ С ПОМОЩЬЮ VBA

Цель

1. Изучить порядок формирования запросов с использованием SQL и VBA в Microsoft Access;
2. Получить практические навыки формирования сложных запросов с использованием языка SQL и языка программирования VBA;

Учебные вопросы

- 1.1. Создание формы для импорта данных из таблицы другой БД;
- 1.2. Построение процедуры для обработки данных.

Литература для подготовки к занятию:

1. Иллюстрированный самоучитель по Access.
<http://www.selfteachers.ru>
2. VBA. Краткое руководство.
<https://coderlessons.com/tutorials/kompiuternoe-programmirovanie/vyuchit-vba/vba-kratkoe-rukovodstvo>
3. Слепцова Л.Д. Программирование на VBA в Microsoft Office 2010. - М.: ООО "И.Д.Вильмс", 2010. - 432 с.
4. Access. Программирование на VBA.
<http://msa.polarcom.ru/st/s0000378.htm>
5. И.С. Осетрова, Н.А. Осипов. Microsoft Visual Basic for Application. Учебное пособие. - СПб: НИУИТМО, 2013. – 120с.
6. Кузин А.В. Базы данных: учеб. — М. : Издательский центр "Академия", 2012. – 320 с.

Содержание отчета

1. Название работы;
2. Для каждого из заданий: название задания и материалы в объеме, указанном в задании.

Вопросы для подготовки к занятию

1. Дать определение макроса. Какими возможностями обладают макросы?
2. Дать определение модуля. Какими возможностями обладают модули?
3. Описать технологию создания процедур на VBA.
4. Описать технологию создания баз данных на VBA.
5. Описать технологию создания таблиц на VBA.
6. Описать технологию создания процедуры обработки событий на VBA.

Актуальность занятия

Системы управления базами данных осуществляют не только стандартные процедуры извлечения и обработки данных, но и должны быть способны адаптироваться под конкретные запросы и требования пользователей. Далеко не всегда они могут обеспечены имеющимися в пакетах стандартными средствами. Поэтому для разработчика необходимо уметь использовать средства извлечения и обработки данных, основанные на индивидуальном программировании необходимых процедур.

Visual Basic for Applications (VBA, Visual Basic для приложений) — много упрощённая реализация языка программирования Visual Basic, встроенная в линейку продуктов Microsoft Office (включая версии для Mac OS), а также во многие другие программные пакеты, такие как AutoCAD, SolidWorks, CorelDRAW, WordPerfect. VBA включает и расширяет функциональность ранее использовавшихся специализированных макро-языков.

VBA является интерпретирующим языком. Он близок к Visual Basic. VBA, будучи языком, построенным на COM, позволяет использовать все доступные в операционной системе COM объекты и компоненты ActiveX. По сути, возможно создание приложения на основе Microsoft Word VBA, использующего только средства Corel Draw.

Написание программ на VBA позволяет:

- сделать базу данных "динамичной";
- уменьшить размер базы данных (программирование позволит не создавать дополнительные объекты);
- обращаться к архивным данным в другие файлы или другие базы данных (т.е. подставлять в форму или в отчет такие данные).
- решать такие задачи, которые стандартными средствами MS Access решить невозможно.

Различают процедуры-подпрограммы и процедуры-функции.

Процедуры-подпрограммы (Sub) не возвращают значения вызывающей процедуре.

Процедура-подпрограмма может выполнять любые действия, например корректировать данные базы, выполнять вычисления в полях, открывать формы, печатать отчеты.

Формат процедуры-подпрограммы имеет следующий вид:

```
[Private!Public] [Static] Sub <Имя процедуры>([список аргументов])
[<описание переменных>]
[<инструкции>]
[Exit Sub]
[<инструкции>]
```

End Sub

Здесь Sub, End — отмечают соответственно начало и конец тела процедуры;

Public — указывает, что процедура Sub является общей, т.е. доступной для всех других процедур во всех модулях;

Private — указывает, что процедура Sub доступна только для процедур того модуля, в котором она описана;

Static — указывает, что значения локальных переменных процедуры сохраняются между вызовами этой процедуры; Exit Sub — приводит к немедленному завершению процедуры Sub.

Процедура-функция (Function) возвращает значение, которое присваивается ее имени внутри процедуры.

Формат процедуры-функции имеет следующий вид:

```
[Private!Public]   [Static]   Function   <Имя   процедуры>[(<список
аргументов>)]
  [<описание переменных>]
  [<инструкции>]
  [<имя процедуры>=<выражение>]
  [Exit Function]
  [<инструкции>]
  [<имя процедуры>=<выражение>]
End Function
```

В теле процедуры-функции (в отличие от процедуры-подпрограммы) присутствует инструкция присваивания имени процедуры значения, вычисляемого выражением. Эта инструкция позволяет возратить значение из процедуры-функции в место ее вызова.

При описании переменных обычно используется инструкция присвоения Dim, которая присваивает выражение переменной или константе. Инструкции присвоения всегда включают в себя знак равенства (=). Для присвоения значения переменной, описанной как объект, применяется инструкция Set.

Задание 1.1. Создание формы для импорта данных из таблицы другой БД

Для созданной на предыдущих занятиях базы данных создать новую форму для импорта данных из таблицы другой БД с использованием SQL запроса и языка программирования VBA. Заготовку формы создать при помощи конструктора форм. Источник и приемник данных подготовить самостоятельно на основе имеющейся Схемы данных.

Последовательность выполнения задания.

1. Создать новую, пустую форму при помощи вкладки меню **Создать-Конструктор форм**.

2. Добавить на нее 3 элемента **Поле** и один элемент **Кнопка** (рис.1.1)

3. Разместите их так, как показано на рис.1.2.

4. Для элемента Надпись1 задать значение «Подпись» равной «Путь к файлу БД»

5. Для элементов Надпись2 и Надпись3 задать значение «Подпись» равной соответственно "Наименование таблицы которую нужно выбрать" и "Наименование новой таблицы"

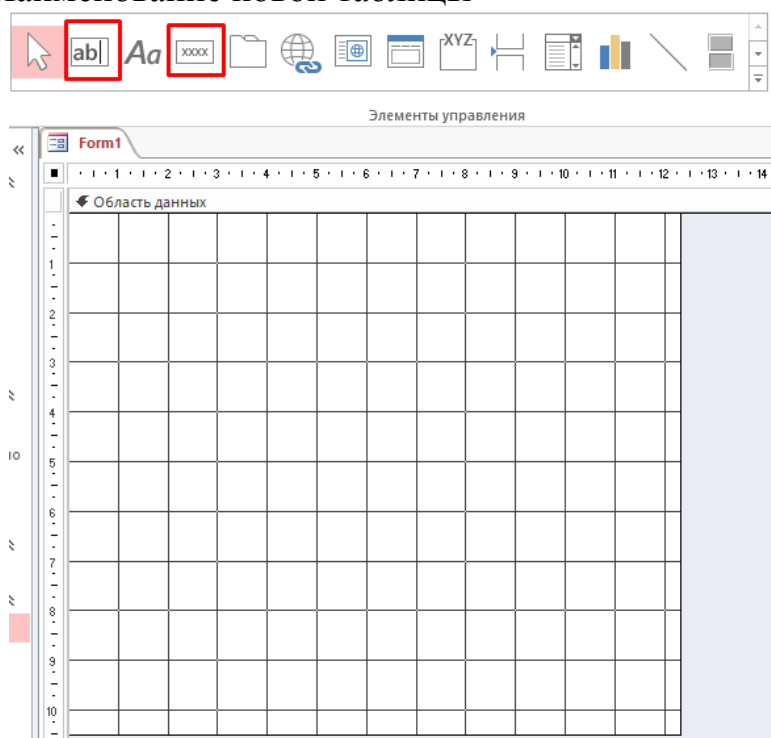


Рис.1.1. Добавление элементов Кнопка и Поле

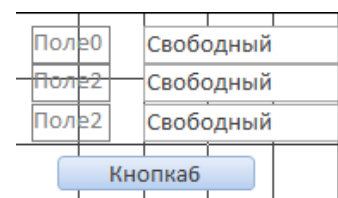


Рис.1.2. Размещение элементов Кнопка и Поле

Путь до файла БД -

Наименование таблицы которую нужно выбрать

Наименование новой таблицы

Рис.1.3. Пример оформления формы

6. Для элемента Кнопка задать значение поля "Подпись" равной "Выполнить импорт". Должна получиться форма подобная рис.1.3.

7. В Окне свойств для кнопки установить в закладке **События** значение в поле **Нажатие кнопки** "Процедура обра-

ботки событий", раскрыв выпадающее меню.

В результате база данных перейдет в режим использования встроенного редактора кода VBA, при этом, откроется окно редактирования кода с кодом следующего содержания:

```
Private Sub Кнопка0_Click()
```

```
...
```

```
End Sub
```

Пояснения операторов в дальнейшем будут показаны комментариями, которые обозначаются апострофом после оператора или с новой строки. Допустимо применять ключевое слово REM в начале текста, но это занимает больше места.

```
Private Sub Кнопка0_Click() 'Объявление о начале выполнения процедуры
                              'по нажатию кнопки
```

```
...
```

```
End Sub                      'Конец процедуры
```

8. Вписать в окно редактирования код для выполнения запроса SQL и сохранить его:

```
Private Sub Кнопка0_Click()
```

```
    'Объявление переменной dbsCurrent типа Database (База Данных)
```

```
Dim dbsCurrent As Database
```

```
    'Объявление переменной qryTestCreate типа QueryDef (Запрос)
```

```
Dim qryTestCreate As QueryDef
```

```
    'Объявление переменной qryTest типа QueryDef (Запрос)
```

```
Dim qryTest As QueryDef
```

```
    'Объявление переменных nname, name, patch типа String (Строка)
```

```
Dim nname, name, patch As String
```

```
    'Присваивание переменным значения полей которые введет пользователь
```

```
nname = Me.Поле3.Value
```

```
name = Me.Поле2.Value
```

```
patch = Me.Поле1.Value
```

```
    'Установка значения переменной dbsCurrent = CurrentDb
```

```
    'т.е. что это переменная работает с нашей текущей БД
```

```
Set dbsCurrent = CurrentDb
```

```
    'Устанавливаем значение переменной qryTestCreate равной созданному
```

```
    'новому запросу (команда на создание нового запроса
```

```
    '«dbsCurrent.CreateQueryDef») где "запрос" – Имя нашего запроса и
```

```
    'далее идет текст запроса в который в нужные места вставляться
```

```
    ' значения наших переменных (вставка производится
```

```
    ' благодаря операторам +)
```



```
Set qryTestCreate = dbsCurrent.CreateQueryDef("запрос", "SELECT *
INTO [" + nname + "] FROM " + name + " IN '" + patch + "';")
```

'Исполнение созданного запроса

```
qryTestCreate.Execute
```

'Удаление ранее созданного запроса

```
dbsCurrent.QueryDefs.Delete ("запрос")
```

```
End Sub
```

9. Закрыть среду разработки, сохранить созданную форму, запустить ее с целью импортирования данных из другой таблицы. Внести данные для импорта в созданную форму и выполнить импорт нажав на кнопку импорта.

10. Проверить корректность выполнения процедуры созданной формы.

В отчете представить:

1. Структуру таблицы, участвующих в обмене.
2. Текст обработчика события.
3. Комментарии строк программы

Задание 1.2. Построение процедуры для обработки данных

1. Написать процедуру VBA, создающую новую базу данных.
2. Написать процедуру, создающую в текущей базе данных таблицу СТУДЕНТЫ с полями Номер студента, ФИО, Предмет 1, Предмет2, Предмет3, Предмет4, Средний балл.
3. Внести в созданную таблицу пять записей во все поля (кроме поля Средний балл).
4. Создать процедуру, подсчитывающую средний балл всех студентов и заносящую рассчитанные значения в поле Средний балл.
5. Создать форму, отображающую данные таблицы СТУДЕНТЫ и содержащую кнопку, запускающую процедуру расчета среднего балла.

Последовательность выполнения задания

1. Открыть новую базу данных и вкладку Модули, написать процедуру, создающую новую базу данных в соответствии с приведенным примером:

'Создание новой базы данных

```
Sub CreateDatabaseX()
```

'Описание переменных

```
Dim myWs As Workspace
```

```
Dim myDb As Database
```

'Определяем стандартный объект Workspace (рабо-

```

'чье пространство)
Set myWs = DBEngine.Workspaces(0)
'Создаем новую базу данных
'с указанным используемым порядком символов dbLangGeneral
Set myDb = myWs.CreateDatabase("C:\NewDB.mdb",
dbLangGeneral)
myDb.Close
End Sub

```

В результате выполнения процедуры на диске C должна появиться новая база данных с названием NewDB.mdb.

2. Написать процедуру, создающую в текущей базе данных таблицу СТУДЕНТЫ с полями Номер студента, ФИО, Предмет 1, Предмет2, Предмет3, Предмет4, Средний балл в соответствии с приведенным примером:

```

'Создание новой таблицы СТУДЕНТЫ в текущей базе данных
Sub CreateTableDefX()
'Определяем переменные
Dim myDb As Database
Dim myTab As TableDef
Dim myF As Field
Set myDb = CurrentDb()
'Создаем новый объект TableDef — таблицу СТУДЕНТЫ
Set myTab = myDb.CreateTableDef("Студенты")
'Создаем новый объект Field — текстовое поле Номер
'студента и добавляем его к семейству полей объекта
'таблицы СТУДЕНТЫ
Set myF = myTab.CreateField("Номер студента",dbInteger)
myTab.Fields.Append myF
'Создаем новый объект Field — текстовое поле ФИО
'и добавляем его к семейству полей объекта таблицы
'СТУДЕНТЫ
Set myF = myTab.CreateField(ФИО, dbText)
myTab.Fields.Append myF
'Создаем новый объект Field — поле Предмет1 и
'добавляем его к семейству полей объекта таблицы
'СТУДЕНТЫ
Set myF = myTab.CreateField("Предмет1", dbInteger)
myTab.Fields.Append myF
'Аналогично поступаем с другими полями таблицы
Set myF = myTab.CreateField("Предмет2", dbInteger)
myTab.Fields.Append myF
Set myF = myTab.CreateField("Предмет3", dbInteger)
myTab.Fields.Append myF

```

```

Set myF = myTab.CreateField("Предмет4", dbInteger)
myTab.Fields.Append myF
Set myF = myTab.CreateField("Средний балл", dbDouble)
myTab.Fields.Append myF
    'Добавляем объект таблицу СТУДЕНТЫ к семейству таб-
    'лиц базы данных
myDb.TableDefs.Append myTab
End Sub

```

3. Открыть созданную таблицу и внести пять записей во все поля (кроме поля Средний балл).

4. Создать процедуру, подсчитывающую средний балл всех студентов и заносящую рассчитанные значения в поле Средний балл в соответствии с приведенным примером:

```

Private Sub SB ()
Dim myDb As Database      'объектная переменная типа базы данных
Dim myRec As Recordset    'объектная переменная типа 'набора записей
Dim sb As Double          'переменная для вычисления 'среднего балла
Dim i As Integer          'переменная цикла
Dim max As Integer        'переменная для хранения числа записей в таблице
Set myDb = CurrentDb()    'Работаем с текущей базой 'данных
    'Открываем набор записей таблицы СТУДЕНТЫ и присваиваем
    'ссылку на него объектной переменной myRec
Set myRec = myDb.OpenRecordset("Студенты")
i = 0
myRec.MoveLast            'Идем к последней записи таблицы
max = myRec.RecordCount    При этом RecordCount содержит число
    ' записей в таблице, которое нужно нам для
    ' вычисления среднего бала в каждой
    'строке таблицы
myRec.MoveFirst           'Переходим к первой записи
    'таблицы и вычисляем средний балл

Do While i < max
sb я (myRec!Предмет1 + myRec!Предмет2 + myRec!Предмет3 +
+ myRec!Предмет4)/4
myRec.Edit                'Заносим значение среднего
    'балла в одноименное поле
myRec![Средний балл] = sb
myRec.Update              'Для внесения данных в поля таблицы
    ' обязательно используются команды Edit и
    'Update.
myRec .MoveNext           'Переходим к следующей записи таблицы и
    'повторяем все пока не достигнем послед-

```

```

i = i + 1           'ней записи.
Loop               'Закрываем набор записей.

myRec.Close
End Sub

```

5. Создать форму, отображающую данные таблицы СТУДЕНТЫ и содержащую кнопку, запускающую процедуру расчета среднего балла.

В отчете представить:

1. Текст процедуры для создания БД;
2. Работающий файл базы данных.

Вопросы для самопроверки

1. Дать определение макроса. Какими возможностями обладают макросы?
2. Дать определение модуля. Какими возможностями обладают модули?
3. Описать технологию создания процедур на VBA.
4. Описать технологию создания баз данных на VBA.
5. Описать технологию создания таблиц на VBA.
6. Описать технологию создания процедуры обработки событий на VBA.
7. Описать технологию создания процедуры обработки событий на VBA.
8. Написать процедуру создания формы на VBA.
9. Написать процедуру создания отчета на VBA.
10. Описать основные свойства форм, доступные при программировании на VBA.
11. Как настроить кнопку на форме, что бы при нажатии на нее выполнялся код VBA?
12. Как в VBA объявить переменную “Data” типа “String”?
13. Как в VBA переменной “patch” присвоить значение поля “Поле5”?

2. РАБОТА С БАЗОЙ ДАННЫХ НА ЯЗЫКЕ SQL

Цель

1. Выработать первичные навыки использования языка SQL для создания и использования баз данных;
2. Получить навыки работы с основными элементами СУБД при создании и редактировании таблиц БД.

Учебные вопросы

- 2.1. Создание и редактирование таблиц;
- 2.2. Разработка запросов к БД.

Литература

1. Введение в язык SQL приложения Access.
<https://support.office.com/ru-ru>
2. Астахова И.Ф. и др. SQL в примерах и задачах. Учебное пособие. - Мн.: Новое знание, 2002. - 176 с.
3. Баканов В.М. Введение в язык запросов SQL запросы к базам данных. Учебное пособие. -М.: МГАПИ, 2002. -61 с.

Содержание отчета

1. Название работы;
2. Для каждого из заданий: название задания и материалы в объеме, указанном в задании.

Вопросы для подготовки к занятию

1. Назначение и основные конструкции операторов SQL;
2. Хара

Актуальность занятия

Системы управления базами данных осуществляют не только стандартные процедуры извлечения и обработки данных, но и должны быть способны адаптироваться под конкретные запросы и требования пользователей. Далеко не всегда они могут обеспечены имеющимися в пакетах стандартными средствами. Поэтому для разработчика необходимо уметь использовать средства извлечения и обработки данных, основанные на индивидуальном программировании необходимых процедур.

Язык SQL (Structured Query Language – язык структурированных запросов) декларативного типа для простого и эффективное извлечение данных из БД. Содержит менее 30 типов предложений. Язык ориентирован на результат, но не создание эффективной процедуры получения результата. В нем нет средств для сложных приложений, он используется совместно с языками программирования.

Структура языка представлена на рис.2.1.

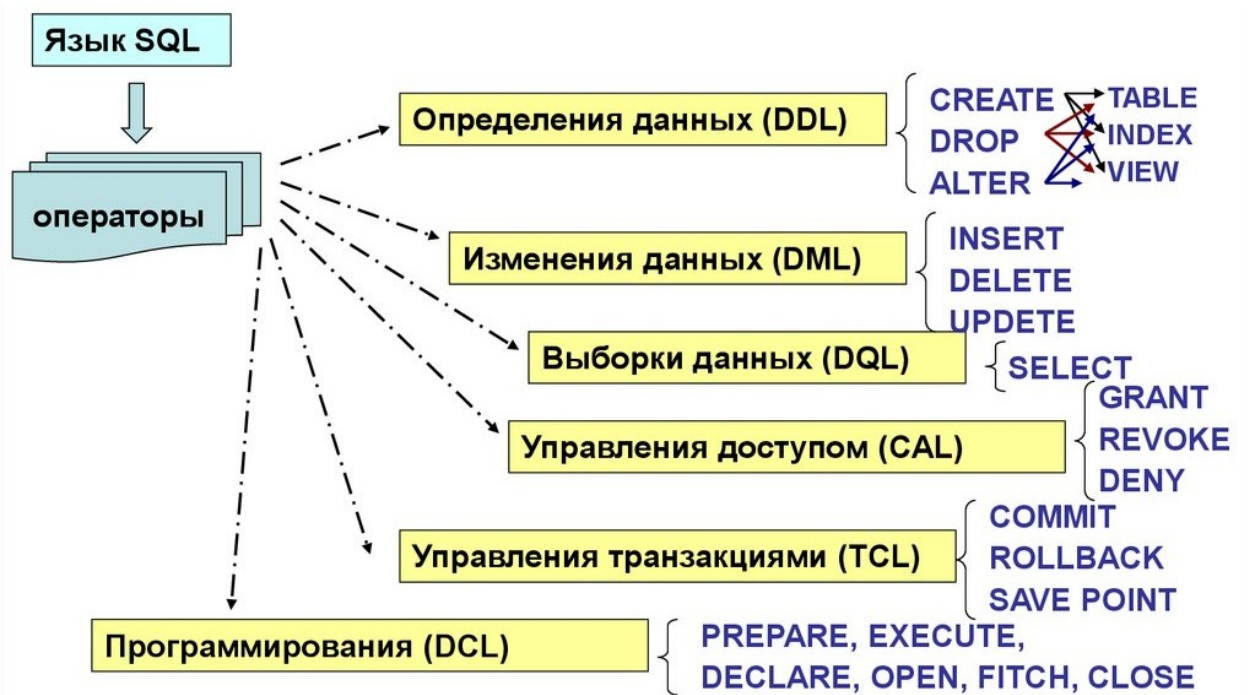


Рис.2.1. Структура языка SQL

Доступ к режиму создания SQL запросов осуществляется через меню Вид в соответствии с рис.2.2.

Оператор SELECT

Формат основных ключевых слов для оператора SELECT:

SELECT [список выбираемых полей]
FROM [таблица] из (таблицы источника)
WHERE [условие] где (условие отбора)
ORDER BY [поле] сортировать по (полю)
 ASC по возрастанию
 DESC по убыванию
GROUP BY [группируемое поле]
AS [новое имя выходного поля] как
TOP N выдать верхние N строк
TOP N PERCENT выдать верхние N процентов строк
OFFSET N пропустить N строк (смещение от начала)

Примеры записи оператора для БД успеваемости студентов:

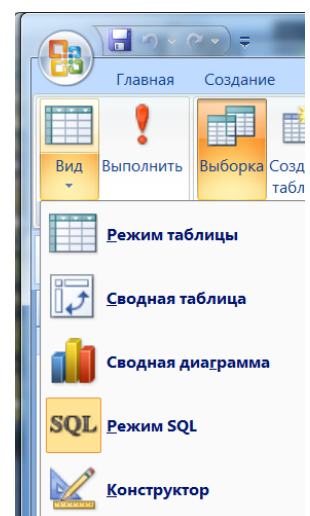


Рис.2.2. Переход в режим SQL

1. Оператор без условий для выдачи первых 10 строк:

```
SELECT TOP 10 Фамилия, Имя, год_рождения
FROM ТаблСотрудники;
```

2. Оператор выбора всех полей с сортировкой по фамилии по возрастанию, затем по имени по убыванию:

```
SELECT * FROM ТаблСотрудники
ORDER BY Фамилия ASC, Имя DESC;
```

3. Выборка фамилий по условию наличия конкретных оценок по двум дисциплинам:

```
SELECT Фамилия, Имя FROM ТаблСтуденты
WHERE Математика>=4 AND Информатика>= 4;
```

4. Запрос с параметром, который вводится уже при выполнении, причем допускается ввод фамилии не полностью:

```
SELECT Фамилия, Имя, (Year(Now())- Year(ТаблСтуденты!ДатаРожд))
AS Возраст
FROM ТаблСтуденты
WHERE ТаблСтуденты.Фамилия
LIKE ("*"+[Введите фамилию]+"*");
```

5. Итоговый запрос (с группировкой):

```
SELECT COUNT (Kod_sotr) AS "Численность"
FROM Sotrudniki
GROUP BY Kod_magazin.
```

Операторы создания таблиц

Основные операторы создания таблиц:

```
CREAT TABLE [Имя таблицы] создать таблицу
DROP TABLE [Имя таблицы] удалить таблицу
ALTER TABLE [Имя таблицы] изменить таблицу
EXEC переименовать таблицу (в Access не работает).
```

Для оператора создания таблицы **CREAT TABLE** применяются атрибуты:

```
PRIMARY KEY первичный ключ
IDENTITY (идентификатор, INCR) – COUNTER
UNIQUE уникальное значение
REFERENCES (ссылка) на внешнюю таблицу
FOREIGN KEY вторичный ключ
CONSTRAINT (ограничение) задает имя ограничения
```

Для оператора создания таблицы **CREATE TABLE** применяются ограничения:

NULL (по умолчанию) пустая строка (**NOT NULL**)

DEFAULT по умолчанию (**В Access не используется**)

CHECK (проверить) ограничения для диапазона (**В Access не используется**)

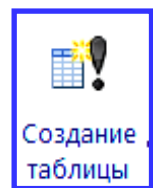
ON DELETE действия при удалении

ON UPDATE действия при изменении

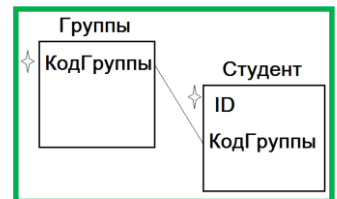
CASCADE, NO ACTION, SET NULL, SET DEFAULT

Примеры записи оператора:

1. Создать таблицу: **CREATE TABLE Студент**
(ID COUNTER PRIMARY KEY,
ФИО CHAR(15) NOT NULL,
Возраст INT DEFAULT 18
CHECK(Возраст >0 AND Возраст <100));



2. Создать связанные таблицы, отражающие структуру:



CREATE TABLE Студент
(ID COUNTER CONSTRAINT PrimaryKey PRIMARY KEY,
ФИО CHAR(15) NOT NULL,
КодГруппы INT REFERENCES Группы(КодГруппы)
ON DELETE CASCADE,
Возраст INT DEFAULT 18
CHECK(Возраст >0 AND Возраст <100));

Возможен альтернативный вариант для задания поля КодГруппы:

КодГруппы int,

.....

FOREIGN KEY (КодГруппы)
REFERENCES Группы (КодГруппы));

3. Удалить таблицу:

DROP TABLE Студент;

4. Изменить таблицу добавление поля:

ALTER TABLE Студенты
ADD НовИмя INT NULL;

Операторы изменения таблицы

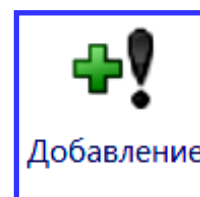
Ключевые слова для выполнения редактирования:

INSERT INTO [поля] **вставить данные в...**
DELETE FROM [Имя таблицы] **удалить записи**
UPDATE [Имя таблицы] **изменить таблицу**
VALUES [значения полей] **назначить константы**
SET [поле=значение] **установить новое значение**

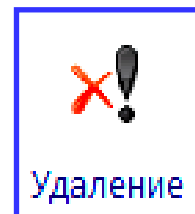
Примеры использования операторов редактирования таблиц

1. Внесение новой записи в таблицу: **INSERT INTO**
Студент

VALUES ('Коротков', 5, 2700);

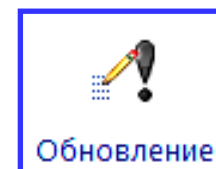


2. Две новые записи в порядке следования полей:
INSERT INTO Товары(Имя, Количество, Цена)
VALUES ('Фото1', 5, 2700),
 ('Фото2', 7, 3000);



3. Удаление записи:

DELETE *
FROM Студент
WHERE Студент.Фамилия = 'Коротков';



4. Обновление полей (замена фамилии):

UPDATE Студент
SET Фамилия='Новый'
WHERE Фамилия='Коротков';

Задание 2.1. Создание и редактирование таблиц

Выполнить средствами языка SQL:

1. Создать 3 связанные таблицы:

Группы (Ключ, Код группы): ДВ-11, ДВ-21, ДВ-31, ДВ-41;

Студенты (Ключ, Фамилия); внести в таблицу 6 записей;

Экзамены (Код, Дисциплина, Дата, Оценка); внести в таблицу 2 экзамена ДВ-21 по 2 человека; 1 экзамен ДВ-31 2 человека;

2. Добавить в таблицу Студенты поле "Имя";

3. Изменить фамилию студентке в таблице "Студенты".

В отчете представить:

1. Коды выражений для каждого задания;
2. Файл с разработанными таблицами.

Задание 2.2. Разработка запросов к БД

Выполнить средствами языка SQL следующие запросы:

1. Алфавитный список хорошистов;
2. Алфавитный список отличников за период (год);
3. Количество успевающих;
4. Количество студентов в учебных группах;
5. Внести оценки по двум экзаменам;
6. Тройку студенту исправить на хорошо;
7. Внести передачу 1 человеку.
8. Средний балл по дисциплинам

В отчете представить:

1. Коды выражений для каждого задания;
2. Файл с разработанными таблицами.

Контрольные вопросы

1. Какие задачи реализуются с помощью языка SQL?
2. В чем заключается назначение механизма ODBC?
3. Каков формат инструкции SELECT? Сформулируйте назначение параметров этой инструкции?
4. Назовите типы данных SQL?
5. Напишите оператор выбора заданных полей из заданной таблицы.
6. Напишите запрос с использованием предикатов? Какие операторы сравнения могут быть использованы?
7. Какие операции над множествами может реализовать оператор SELECT?
8. В чем отличие предложений WHERE и HAVING?
9. Напишите запрос добавления записей в таблицу.
10. Напишите запрос на изменение данных в таблице.
11. Сформируйте запрос на удаление данных в указанной таблице.