**Практическая работа №12**

**Наименование работы:** Проектирование базы данных «Фирма». Создание таблиц, проектирование связей между таблицами. Создание форм для ввода данных, главной кнопочной формы. Работа с формами.

**Цель:** Закрепить основные принципы и приемы работы в СУБД

**Порядок работы**

**Знакомство с Access. Создание таблиц**

*База данных (БД)* – упорядоченная совокупность данных, предназначенных для хранения, накопления и обработки с помощью ЭВМ. Для создания и ведения баз данных (их обновления, обеспечения доступа по запросам и выдачи данных по ним пользователю) используется набор языковых и программных средств, называемых *системой управления базами данных (СУБД).*

Таблица 1 – Типы данных

|  |  |
| --- | --- |
| **Тип**   | **Описание** |
| Текстовый  | Используется для хранения символьных или числовых данных, не требующих вычислений. В свойстве *Размер поля* задается максимальное количество символов, которые могут быть введены в данное поле. По умолчанию размер устанавливается в 50 знаков. Максимальное количество символов, которые могут содержаться в текстовом поле, – 255  |
| Числовой  | Предназначен для хранения числовых данных, используемых в математических расчетах. На вкладках *Общие* и *Подстановка* можно установить свойства числового поля, среди которых *Размер поля, Формат поля, Число десятичных знаков*  |
| Дата/Время  | Используется для представления даты и времени. Выбор конкретного формата даты или времени устанавливается в свойстве *Формат даты*  |
| Денежный  | Предназначен для хранения данных, точность представления которых колеблется от 1 до 4 знаков после запятой. Целая часть может содержать до 15 десятичных знаков  |
| Счетчик  | Предназначен для автоматической вставки уникальных последовательных (увеличивающихся на 1) или случайных чисел в качестве номера новой записи. Номер, присвоенный записи, не может быть удален или изменен. Поля с этим типом данных используются в качестве ключевых полей таблицы  |
| Логический  | Предназначен для хранения одного из двух значений, интерпретируемых как «Да / Нет», «Истина / Ложь», «Вкл. / Выкл.»  |
| Поле объекта OLE  | Содержит данные, созданные в других программах, которые используют протокол OLE. Это могут быть, например, документы Word, электронные таблицы Excel, рисунки, звуковые и видеозаписи и др. Объекты OLE связываются с базой данных Access или внедряются в нее. Сортировать, группировать и индексировать поля объектов OLE нельзя  |
| Гиперссылка  | Специальный тип, предназначенный для хранения гиперссылок  |
| Мастер подстановок  | Предназначен для автоматического определения поля. С его помощью будет создано поле со списком, из которого можно выбирать данные, содержащиеся в другой таблице или в наборе постоянных значений  |

**Объекты базы данных Access**

К объектам базы данных Access относятся:

1. *Таблицы* – предназначены для упорядоченного хранения данных.
2. *Запросы* – предназначены для поиска, извлечения данных и выполнения вычислений.
3. *Формы* – предназначены для удобного просмотра, изменения и добавления данных в таблицах.
4. *Отчеты* – используются для анализа и печати данных.
5. *Страницы доступа к данным* – предназначены для просмотра, ввода, обновления и анализа данных через сеть или из любого места компьютера.
6. *Макросы* – используются для выполнения часто встречающегося набора макрокоманд, осуществляющих обработку данных.
7. *Модули* – предназначены для описания инструкций и процедур на языке VBA.

Основным объектом базы данных является таблица, которая состоит из записей (строк) и полей (столбцов). На пересечении записи и поля образуется ячейка, в которой содержатся данные.

Каждому полю таблицы присваивается *уникальное имя*, которое не может содержать более 64 символов. В каждом поле содержатся данные одного типа.

 Наиболее важные операторы, которые входят в стандарт ANSI SQL, приведены в таблице 1

**Таблица 1** – Операторы доступа к данным

|  |  |  |
| --- | --- | --- |
| **Оператор**  |  | **Действие**  |
|  | **Оператор выбора SELECT**  |
| **SELECT**  |  | Оператор, формирующий в соответствии с SQL-запросом результирующее отношение  |
|  | **Операторы определения данных**  |
| CREATE TABLE  |  | Создает новую  |
| DROP TABLE  |  | Удаляет таблицу из базы данных  |
| ALTER TABLE  |  | Изменяет структуру существующей таблицы  |
| CREATE VIEW  |  | Создает виртуальную таблицу, соответствующую SQLзапросу  |
| DROP VIEW  |  | Удаляет ранее созданное представление  |
| ALTER VIEW  |  | Изменяет ранее созданное представление  |
| CREATE INDEX  |  | Создает индекс для обеспечения быстрого доступа  |
| DROP INDEX  |  | Удаляет ранее созданный индекс  |
| DELETE  |  | Удаляет в соответствии с условиями фильтрации строки из таблицы  |
| INSERT  |  | Вставляет строку в базовую таблицу  |
| UPDATE  |  | Обновляет в соответствии с условиями значения одного или нескольких столбцов  |
|  | **Операторы администрирования данных**  |
| ALTER PASSWORD  |  | Изменить пароль для доступа к базе данных  |
| CREATE DATEBASE  |  | Создать новую базу данных  |
| DROP DATEBASE  |  | Удалить существующую базу данных  |
| GRANT  |  | Предоставить права доступа на ряд действий с объектом базы данных  |
| REVOKE  |  | Лишить прав доступа к некоторому объекту  |

**Оператор выбора SELECT**

Одним из основных инструментов обработки данных в СУБД является выборка данных с помощью запросов. Запрос строится на основе одной или нескольких таблиц. Запрос позволяет выбрать необходимые данные из одной или нескольких таблиц, произвести вычисления и получить результат в виде таблицы. Через запрос можно производить обновление данных в таблицах, создать новую таблицу, используя данные из существующих объектов базы данных. Результатом запроса на выборку является таблица.

Ключевое слово **SELECT** сообщает СУБД, что эта команда - запрос. Все запросы начинаются этим словом с последующим пробелом.

Синтаксис оператора (команды) выбора **SELECT** имеет вид:

**SELECT**[ALL / DISTINCT] (<список полей>/\*)

**FROM** (<список таблиц>)

**[WHERE** <предикат - условие выборки или соединения>]

[**GROUP BY**< список полей группировки>]

[**HEVING** < предикат - условие для группы>]

[**ORDER BY** < список полей, определяющих порядок сортировки>]

Звездочка в операторе **SELECT** означает выбор всех столбцов из таблицы.

1. Создадим базу данных «Фирма». Сотрудники данной организации работают с клиентами и выполняют их заказы.

Если все сведения поместить в одной таблице, то она станет очень неудобной для работы. В ней начнутся повторы данных. Всякий раз, когда сотрудник Иванов будет работать с какой-либо фирмой, придется прописывать данные о сотруднике и клиенте заново, в результате чего можно допустить множество ошибок. Чтобы уменьшить число ошибок, можно исходную таблицу разбить на несколько таблиц и установить связи между ними. Это будет более рационально, чем прежде.

Таким образом, необходимо создать 3 таблицы: *Сотрудники*, *Клиенты* и *Заказы.*

**Сотрудники**

|  |  |
| --- | --- |
| **Имя поля**   | **Тип данных**   |
| Код сотрудника  | Счетчик  |
| Фамилия  | Текстовый  |
| Имя  | Текстовый  |
| Отчество  | Текстовый  |
| Должность  | Текстовый  |
| Телефон  | Текстовый  |
| Адрес  | Текстовый  |
| Дата рождения  | Дата/Время  |
| Заработная плата  | Денежный  |
| Фото  | Объект OLE  |
| Эл\_почта  | Гиперссылка  |

**Клиенты**

|  |  |
| --- | --- |
| **Имя поля**   | **Тип данных**  |
| Код клиента  | Счетчик  |
| Название компании  | Текстовый  |
| Адрес  | Текстовый  |
| Номер телефона  | Текстовый  |
| Факс  | Числовой  |
| Адрес электронной почты  | Гиперссылка  |
| Заметки  | Поле МЕМО  |

**Заказы**

|  |  |
| --- | --- |
| **Имя поля**   | **Тип данных**  |
| Код заказа  | Счетчик  |
| Код клиента  | Числовой  |
| Код сотрудника  | Числовой  |
| Дата размещения  | Дата/Время  |
| Дата исполнения  | Дата/Время  |
| Сумма  | Денежный  |
| Отметка о выполнении  | Логический  |

1. Отдельные таблицы, содержащие информацию по определенной теме, необходимо связать в единую структуру базы данных. Для связывания таблиц следует задать *ключевые поля*. Ключ состоит из одного или нескольких полей, значения которых *однозначно* определяют каждую запись в таблице. Наиболее подходящим в качестве ключевого поля является *«Счетчик»*, так как значения в данном поле являются уникальными (т. е. исключают повторы).
2. Откройте таблицу *Сотрудники* в режиме Конструктора.
3. Нажмите правой кнопкой мыши на поле *Код сотрудника* и в появившемся контекстном меню выберите команду *Ключевое поле*. Если в таблице необходимо установить несколько ключевых полей, то выделить их можно, удерживая клавишу *Ctrl*.
4. Для таблицы *Клиенты* установите ключевое поле *Код клиента*, а для таблицы *Заказы – Код заказа*.
5. Таблица *Заказы* содержит поля *Код сотрудника* и *Код клиента*. При их заполнении могут возникнуть некоторые трудности, так как не всегда удается запомнить все предприятия, с которыми работает фирма, и всех сотрудников с номером кода. Для удобства можно создать раскрывающиеся списки с помощью *Мастера подстановок*.
6. Откройте таблицу *Заказы* в режиме Конструктора.
7. Для поля *Код сотрудника* выберите тип данных *Мастер подстановок.*
8. В появившемся окне выберите команду «Объект «*столбец подстановки» будет использовать значения из таблицы или запроса*» и щелкните на кнопке *Далее*.
9. В списке таблиц выберите таблицу *Сотрудники* и щелкните на кнопке *Далее*.
10. В списке *Доступные поля* выберите поле *Код сотрудника* и щелкните на кнопке со стрелкой, чтобы ввести поле в список *Выбранные поля*. Таким же образом добавьте поля *Фамилия* и *Имя* и щелкните на кнопке *Далее*.
11. Выберите порядок сортировки списка по полю *Фамилия.*
12. В следующем диалоговом окне задайте необходимую ширину столбцов раскрывающегося списка.
13. Установите флажок *Скрыть ключевой столбе»* и нажмите кнопку *Далее*.
14. На последнем шаге *Мастера подстановок* замените при необходимости надпись для поля подстановок и щелкните на кнопке *Готово*.
15. Аналогичным образом создайте раскрывающийся список для поля *Код клиента*.
16. После создания ключевых полей можно приступить к созданию связей.

Существует несколько типов отношений между таблицами:

1. при отношении «*один-к-одному»* каждой записи ключевого поля в первой таблице соответствует только одна запись в связанном поле другой таблицы, и наоборот. Отношения такого типа используются не очень часто. Иногда их можно использовать для разделения таблиц, содержащих много полей, для отделения части таблицы по соображениям безопасности;
2. при отношении *«один-ко-многим»* каждой записи в первой таблице соответствует несколько записей во второй, но запись во второй таблице не может иметь более одной связанной записи в первой таблице;
3. при отношении *«многие-ко-многим»* одной записи в первой таблице могут соответствовать несколько записей во второй таблице, а одной записи во второй таблице могут соответствовать несколько записей в первой.
4. Закройте все открытые таблицы, так как создавать или изменять связи между открытыми таблицами нельзя.
5. Выполните команду: вкладка ленты *Работа с базами данных→* кнопка .
6. Если ранее никаких связей между таблицами базы не было, то при открытии окна *Схема данных* одновременно открывается окно *Добавление таблицы*, в котором выберите таблицы *Сотрудники, Клиенты* и *Заказы.*
7. Если связи между таблицами уже были заданы, то для добавления в схему данных новой таблицы щелкните правой кнопкой мыши на схеме данных и в контекстном меню выберите пункт *Добавить таблицу*.
8. Установите связь между таблицами *Сотрудники* и *Заказы*, для этого выберите поле *Код сотрудника* в таблице *Сотрудники* и перенесите его на соответствующее поле в таблице *Заказы.*
9. После перетаскивания откроется диалоговое окно *Изменение связей* (рисунок 3), в котором включите флажок *Обеспечение условия целостности*. Это позволит предотвратить случаи удаления записей из одной таблицы, при которых связанные с ними данные других таблиц останутся без связи.



Рисунок 3 – Создание связи между таблицами

1. Флажки *Каскадное обновление связанных полей* и *Каскадное удаление связанных записей* обеспечивают одновременное обновление или удаление данных во всех подчиненных таблицах при их изменении в главной таблице.
2. Параметры связи можно изменить, нажав на кнопку *Объединение*.
3. После установления всех необходимых параметров нажмите кнопку *ОК*.
4. Связь между таблицами *Клиенты* и *Заказы* установите самостоятельно.
5. В результате должна получиться схема данных, представленная на рисунке 4.



Рисунок 4 – Схема данных

В приведенном примере используются связи «один-ко-многим». На схеме данных они отображаются в виде соединительных линий со специальными значками около таблиц. Связь «один-ко-многим» помечается «1» вблизи главной таблицы (имеющей первичный ключ) и «∞» вблизи подчиненной таблицы (имеющей внешний ключ). Связь «один-к-одному» помечается двумя «1» (оба поля таблиц имеют первичные ключи). Неопределенная связь не имеет никаких знаков. Если установлено объединение, то его направление отмечается стрелкой на конце соединительной линии (ни одно из объединенных полей не является ключевым и не имеет уникального индекса).

1. В таблицу *Сотрудники* внесите данные о семи работниках.
2. В таблицу *Клиенты* внесите данные о десяти предприятиях, с которыми работает данная фирма.
3. В таблице *Заказы* оформите несколько заявок, поступивших на фирму.
4. Покажите работу преподавателю.

**Контрольные вопросы:**

1. С помощью чего можно создавать таблицы?
2. Что такое система управления базами данных (СУБД)?
3. Чем отличается Microsoft Excel от Microsoft Access?
4. Какие объекты базы данных MicrosoftAccess вы знаете?
5. Какой объект в базе данных является основным?
6. Что называется полями и записями в БД?
7. Какие типы данных вы знаете?
8. Как можно переименовать поле?
9. Что такое ключевое поле?
10. Как установить несколько ключевых полей?
11. Как установить связи между таблицами?
12. Какие существуют отношения между таблицами?
13. Что означают на схеме данных «1» и «∞»?
14. Зачем нужен *Мастер подстановок*?