# Основные понятия MS Access

**Система управления базами данных** (**СУБД**) – программа для создания и использования баз данных.

**База данных** (**БД**) – хранилище данных некой предметной области, организованное для удобного накопления, быстрого поиска и обработки данных.

**Объекты базы данных Access** – таблицы, запросы, формы, отчеты, макросы и модули.

**Таблица** – множество строк (записей), содержащих данные, разнесенные по поименованным столбцам (полям).

**Поле** – столбец таблицы. Каждое поле таблицы имеет уникальное имя, характеризуется типом данных и свойствами, зависящими от этого типа.

**Тип данных** – характеристика поля, определяющая тип данных, который может содержать это поле. Существуют следующие типы данных: текстовый, числовой, дата/время, денежный, логический, счетчик, гиперссылка, MEMO, OLE и др.

**Свойства поля** — это набор характеристик, обеспечивающих дополнительные возможности управления хранением, вводом и отображением данных в поле. Перечень доступных свойств зависит от типа данных поля. Например, есть такие свойства, как: Размер поля, Обязательное поле, Условие на значение и др.

**Ключевое поле (Первичный ключ)** – поле с уникальными непустыми значениями, однозначно идентифицирующими каждую запись в таблице. Применяется для связи таблиц.

**Ключевое поле (Первичный ключ)** – одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ должен иметь уникальные непустые значения. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

**Внешний ключ** – поле подчиненной таблицы, которое связано с ключевым полем главной таблицы. Если включено обеспечение целостности, то каждое значение внешнего ключа совпадает с одним из существующих значений в связанном ключевом поле.

**Запись** – строка таблицы. Каждая запись таблицы содержит сведения о каком-либо конкретном объекте предметной области.

**Поле записи** – пересечение строки и столбца (ячейка таблицы).

**Значение поля записи** – содержимое соответствующей ячейки таблицы.

**Поле со списком** – элемент управления, который состоит из поля, в которое можно вводить значения, и раскрывающегося списка заранее подготовленных значений, из которого можно выбрать значение и сохранить его в поле. Использование поля со списком делает более удобной работу с базой данных и снижает вероятность ошибок ввода.

**Свойства подстановки поля со списком**: Свойство «Источник строк» содержит имя таблицы, из которой берутся данные для поля со списком. Свойство «Присоединенный столбец» содержит номер столбца указанной таблицы. Столбцы нумеруются слева направо, начиная с единицы.

**Схема данных** – графическое представление структуры базы данных, содержит списки полей таблиц и межтабличные линии связи. Позволяет устанавливать (удалять) межтабличные связи и изменять параметры связи.

**Связь таблиц.** Обычно в базе данных создается несколько таблиц. В одной таблице хранится информация об объектах одного типа, в другой таблице - об объектах другого типа. Если разнотипные объекты реального мира связаны между собой, то и таблицы могут быть связаны для отражения этой связи. Разнотипные объекты могут иметь общее свойство. Информация об этом свойстве хранится в столбце одной таблицы и в столбце другой таблицы. С помощью этих столбцов и осуществляется межтабличная связь.

**Список полей** – окно с перечнем полей таблицы. Ключевые поля выделены специальным значком.

**Линия связи** – линия, соединяющая ключевое поле главной таблицы и внешний ключ подчиненной таблицы. Если включено обеспечение целостности, то у концов линии появляются символы, показывающие тип связи (1 ко многим, или 1:1).Типы межтабличных связей: "один-к-одному", "один- ко-многим", "многие-ко-многим".

**Связывание таблиц** осуществляется способом перетаскивания первичного ключа главной таблицы на внешний ключ подчиненной таблицы в схеме данных.

**Требования к связываемым полям**: связываемые поля должны иметь одинаковую по смыслу информацию и одинаковый тип данных.

**Главная и подчиненная таблица**: если таблицы связаны связью «один-ко-многим», то таблица на стороне «один» называется главной, а на стороне «многие» – подчиненной таблицей. Первичный ключ главной таблицы связан с внешним ключом подчиненной таблицы. Если таблицы связаны связью «один-к-одному», то левая таблица в окне "Изменение связей" называется главной, а правая - подчиненной (связанной) таблицей.

**Целостность данных** – правила, которые СУБД автоматически соблюдает при вводе и удалении значений в связанных полях таблиц. Обеспечение целостности данных можно включить либо отключить при создании связи в схеме данных. Правила целостности:

1. первичный ключ должен содержать уникальные непустые значения,
2. внешний ключ должен содержать только те значения, которые уже имеются среди значений первичного ключа. Из этого следует:
   * нельзя вводить во внешний ключ значения, которых нет в первичном ключе,
   * нельзя изменять значения первичного ключа, для которых имеются совпадающие

значения во внешнем ключе (если только не разрешено каскадное обновление связанных полей),

* + нельзя удалять записи в главной таблице, для которых имеются подчиненные записи в подчиненной таблице (если только не разрешено каскадное удаление связанных записей).

**Каскадное обновление связанных полей**: для автоматического обновления значений внешнего ключа в подчиненной таблице при изменении значения первичного ключа в главной таблице, установите флажки «Обеспечение целостности данных» и «Каскадное обновление связанных полей». Для предотвращения изменений значения первичного ключа в главной таблице, если существуют связанные записи в подчиненной таблице, установите флажок «Обеспечение целостности данных» и снимите флажок «Каскадное обновление связанных полей».

**Каскадное удаление связанных записей**: для автоматического удаления связанных записей в подчиненной таблице при удалении записи в главной таблице установите флажки «Обеспечение целостности данных» и «Каскадное удаление связанных записей». Для предотвращения удаления записей из главной таблицы, если имеются связанные записи в подчиненной таблице, установите флажок «Обеспечение целостности данных» и снимите флажок «Каскадное удаление связанных записей».

**Фильтр** – набор условий, применяемых для отбора или сортировки записей.

**Условие отбора в фильтре** – выражение, относящееся к определенному полю таблицы, используемое для отбора записей, удовлетворяющих этому выражению.

**Выражение** – формула, записанная с использованием операторов , констант, функций, имен объектов базы данных. В результате вычисления формулы получается единственное значение. Правила, используемые при записи выражений:

* Числа вводятся без ограничителей, например, 21
* Текст заключается в кавычки, например «Иванов»
* Даты ограничиваются символами #, например, #10/01/99#

**Операторы**, используемые в выражениях:

* арифметические: \*, +, -, /,^;
* сравнения: <, <=, >, >=, =, < >;
* логические: And (И), Not (Нет), Or (Или);
* Like — для нахождения части значения поля;
* In - для определения, содержится ли элемент данных в списке значений;
* Between… And - для выбора значений из определенного интервала.

**Форма** – объект базы данных. Различают формы ввода-вывода и управляющие формы. Форма ввода-вывода представляет собой окно, специально разработанное для удобства ввода и просмотра информации. Управляющая форма представляет собой окно, специально разработанное для удобства работы с базой данных.

**Источник данных формы** – таблицы или запросы, на основе которых создается форма. Источником данных формы могут служить несколько таблиц или запросов.

**Свободная форма** – форма, для которой не указан источник данных. Используется для построения управляющих форм.

**Элемент управления**. Объект графического интерфейса пользователя (такой как поле, флажок, полоса прокрутки или кнопка), позволяющий пользователям управлять приложением. Элементы управления используются для отображения данных или параметров, для выполнения действий, либо для упрощения работы с интерфейсом пользователя.

**Присоединенное поле** – текстовое поле в форме, в котором отображается информация из поля текущей записи таблицы или запроса, на основе которых построена форма.

**Свободное поле** – текстовое поле в форме или отчете, которое не присоединено ни к какому источнику данных. В это поле пользователь может вводить свою информацию.

**Запрос** – объект базы данных. Запросы используются для выборки данных из таблиц, для изменения данных в таблицах, для вычислений.

**Запрос на выборку** – средство базы данных, позволяющее выбрать информацию из нескольких таблиц и других запросов в соответствии с условиями отбора. Кроме того, запросы на выборку позволяют производить вычисления.

**Запрос на обновление** – средство обновления устаревших данных в таблице.

**Запрос на добавление** – средство добавления записей из одной таблицу в другую. Таблицы должны иметь поля, с одинаковой по смыслу информацией.

**Запрос на удаление** – средство удаления из таблицы или нескольких таблиц записей, удовлетворяющих заданным условиям.

**Запрос на создание таблицы** – средство создания новой таблицы на основе существующих таблиц.

**Перекрёстный запрос** позволяет произвести вычисления в таблице базы данных и выдать результаты в виде перекрёстной таблицы . Применение перекрёстных таблиц позволяет более компактно и наглядно представить обобщенную информацию, сформированную на основе исходной таблицы базы данных.

**Запрос с параметром** – запрос, после запуска которого запрашивается ввод условий отбора. Чтобы создать запрос с параметром надо в условии отбора набрать в квадратных скобках произвольный поясняющий текст. Запросы с параметром создаются для удобства работы пользователя. Источник данных для запроса – таблицы и другие запросы, включенные в запрос.

**Вычисляемое поле** – поле, определенное в запросе для вычисления выражения для каждой записи с использованием данных из одного или нескольких полей.

**Построитель выражений** – средство, позволяющее автоматизировать построение сложных выражений.

**Отчет** – объект базы данных, обеспечивающий формирование печатных документов на основе информации из базы данных.

**Приложение** – база данных, обеспечивающая удобную среду общения пользователя с базой данных. Это подразумевает работу, в основном, с помощью управляющих форм, без непосредственного обращения к таблицам и запросам.

**Связь между формой и запросом** может быть организована так:

1. В форме создать свободное поле (или поле со списком)
2. В запросе в условии отбора сделать ссылку на это поле
3. В форме сделать кнопку для запуска этого запроса

**Связь между формой и отчетом** можно установить так:

1. Создать отчёт на основе запроса
2. В форме создать свободное поле (или поле со списком)
3. В запросе в условии отбора использовать ссылку на поле (или поле со списком), расположенное в форме
4. В форме создать кнопку для открытия отчета.

# Постановка задачи на разработку базы данных

Наша задача ‒ разработать базу данных для учета семестровой успеваемости студентов.

#### Краткое описание предметной области

В учебном заведении формируются группы студентов. Студенты в каждом семестре сдают зачеты и экзамены по учебным дисциплинам в соответствии с учебным планом. Преподаватели экзаменуют студентов и выставляют оценки в соответствии с перечнем возможных оценок.

#### Этапы решения задачи

1. Исходя из анализа предметной области, надо спроектировать таблицы. В таблицах не должно быть дублирования информации. Информация должна вводиться однократно, а

использоваться многократно.

1. Нам будет нужно создать в таблицах поля, которые позволят организовать связь между таблицами. Эти поля будем создавать с помощью мастера подстановок. Мастер подстановок создает так называемые поля со списком, которые позволяют использовать данные из одних таблиц в связанных с ними таблицах. Использование полей со списком значительно ускоряет ввод данных и сокращает количество возможных ошибок.
2. Далее создадим схему базы данных, которая наглядно представит взаимосвязи таблиц и позволит нам обеспечить целостность базы данных.
3. На основе созданных таблиц будем создавать необходимые запросы.
4. Для удобства работы создадим формы ввода, изменения и просмотра данных.
5. Потребуется также создание печатных отчетов.
6. В заключение создадим управляющую форму, которая объединит все объекты нашей базы в единое целое и предоставит удобный интерфейс людям, работающим с базой данных.