

ИСПОЛЬЗОВАНИЕ СТРОКОВЫХ
ФУНКЦИЙ ПРИ РАБОТЕ
С ЛИНГВИСТИЧЕСКОЙ
КАРТОТЕКОЙ MS ACCESS

Пользуясь справочными материалами этой презентации, решите приведенные ниже задачи (все запросы создавайте в БД **к. задачам.mdb**).

Задача #1

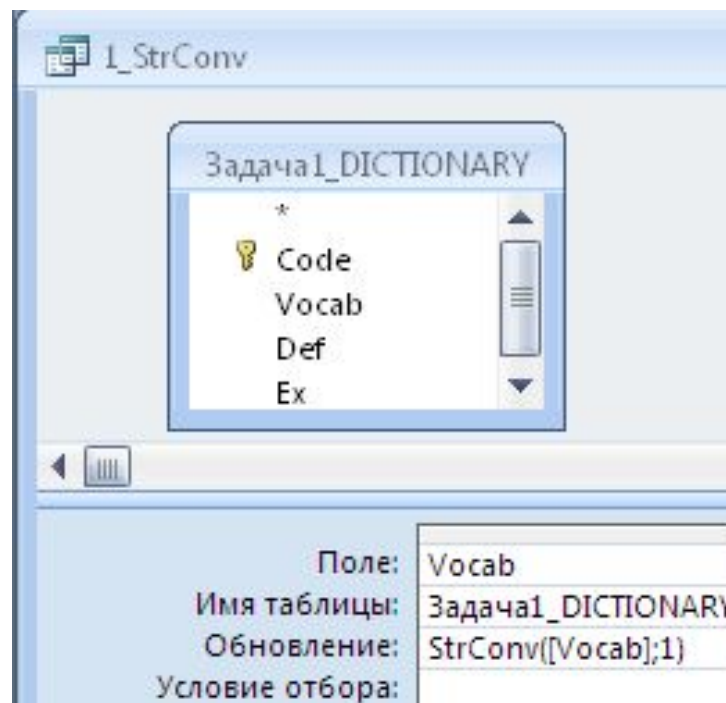
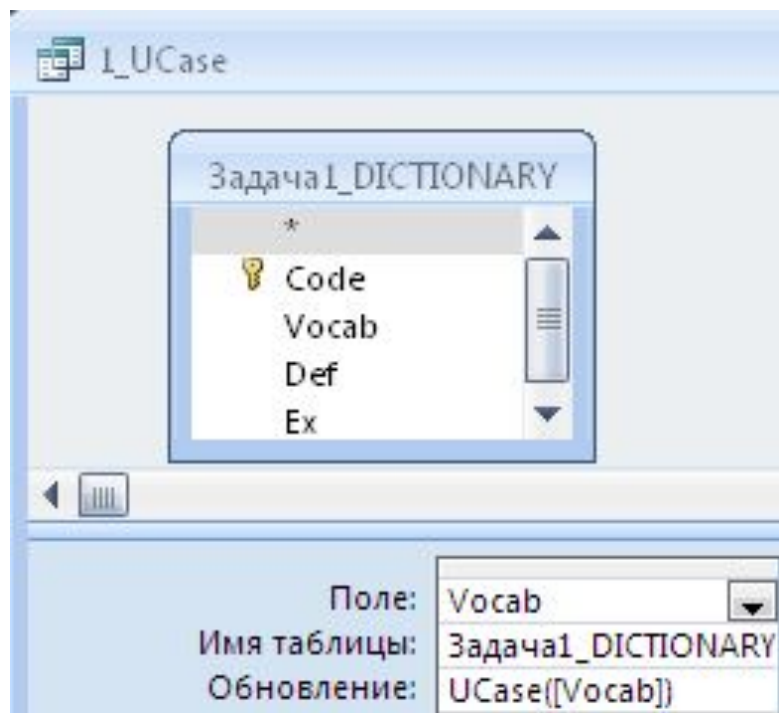
ЗАДАЧА № 1. Дано: поле Vocab таблицы **Задача1_DICTIONARY** (тип данных: текстовый; значение этого поля для каждой записи – заголовочное слово словарной статьи; единый стиль оформления заголовочных слов не выдержан: часть вокабул набрана с первой прописной буквы, часть – полностью прописными буквами, часть – строчными буквами).

Задание: преобразовать записи в поле Vocab так, чтобы все заголовочные слова были набраны прописными буквами (иначе говоря – были набраны в верхнем регистре).

Подсказка: используйте функцию **UCase** или **StrConv**. Тип запроса: **обновление**.

Справку по функциям см. на следующем слайде.

Окна запросов:



Функция UCase

Возвращает заданную строку, преобразованную в верхний регистр.

Синтаксис: **UCase(строка)**

В прописные буквы преобразуются только строчные буквы, а все прописные буквы и прочие знаки остаются без изменения.

Функция StrConv

Возвращает значение типа Variant (String), преобразованное способом, заданным аргументом *тип_преобразования*.

Синтаксис: **StrConv(строка, тип_преобразования [, идентификатор])**

Функция **StrConv** имеет следующие аргументы:

- аргумент **строка** Обязательный аргумент. Преобразуемое строковое выражение.
- аргумент **тип_преобразования** может принимать следующие значения:
 - 1 – для преобразования знаков строки в верхний регистр
 - 2 – для преобразования знаков строки в нижний регистр
 - 3 – для преобразования знаков первой буквы каждого слова строки в прописную.
- ...
- аргумент **[, идентификатор]** – **необязательный**.

Задача #2

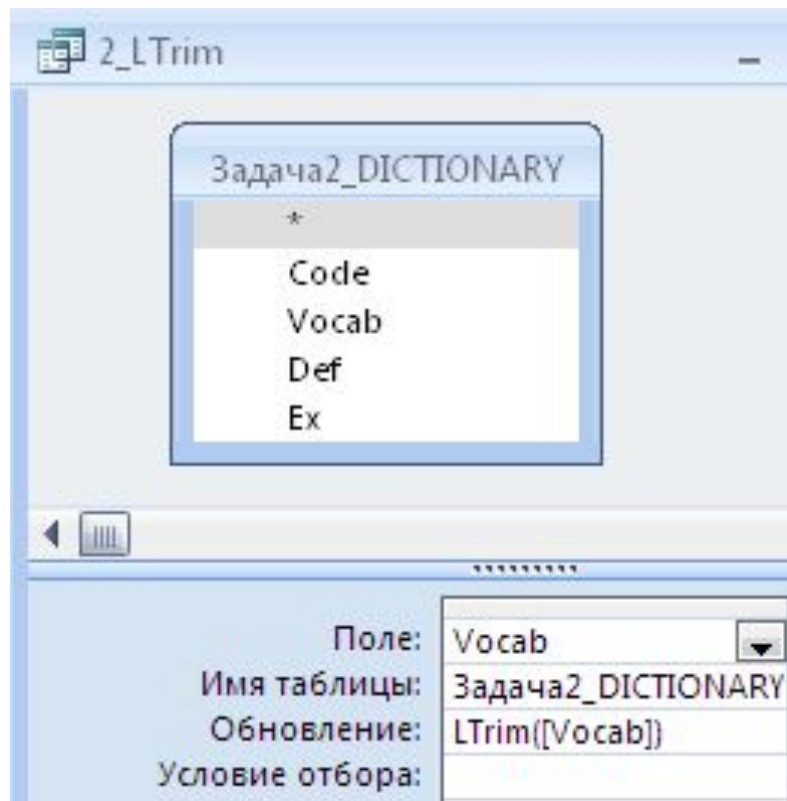
ЗАДАЧА № 2 . Дано: поле Vocab таблицы **Задача2_DICTIONARY** (тип данных: текстовый; значение этого поля для каждой записи – заголовочное слово словарной статьи; единый стиль оформления заголовочных слов не выдержан: в записи встречаются начальные пробелы).

Задание: исключить начальные пробелы в поле Vocab.

Подсказка: используйте функцию **LTrim** . Тип запроса: **обновление**.

Справку по функции см. на следующем слайде.

Окно запроса:



Функции LTrim, RTrim и Trim

Возвращает копию указанной строки без начальных пробелов (функция **LTrim**), конечных пробелов (функция **RTrim**) или начальных и конечных пробелов (функция **Trim**).

Синтаксис

LTrim(*Строка*)

RTrim(*Строка*)

Trim(*Строка*)

Обязательный аргумент **Строка** может содержать любое допустимое строковое выражение. Если аргумент **Строка** имеет значение Null, возвращается **Null**.

Задача #3

ЗАДАЧА № 3. Дано: поле Vocab таблицы **Задача3_DICTIONARY** (тип данных: текстовый; значение этого поля для каждой записи – заголовочное слово словарной статьи).

Задание: получить обратный словарь словоформ (словарь не нужно помещать в отдельную таблицу).

Подсказка: используйте функцию **StrReverse** . Тип запроса: **выборка**.

Справку по функции см. на следующем слайде.

Окно запроса:

Поле:	rev: StrReverse([Vocab])
Vocab	
Имя таблицы: Задача3_DICTIONARY	
Сортировка: по возрастанию	
Вывод на экран: <input checked="" type="checkbox"/>	<input type="checkbox"/>
Условие отбора: Is Not Null	
или:	

StrReverse err if NULL

Функция StrReverse

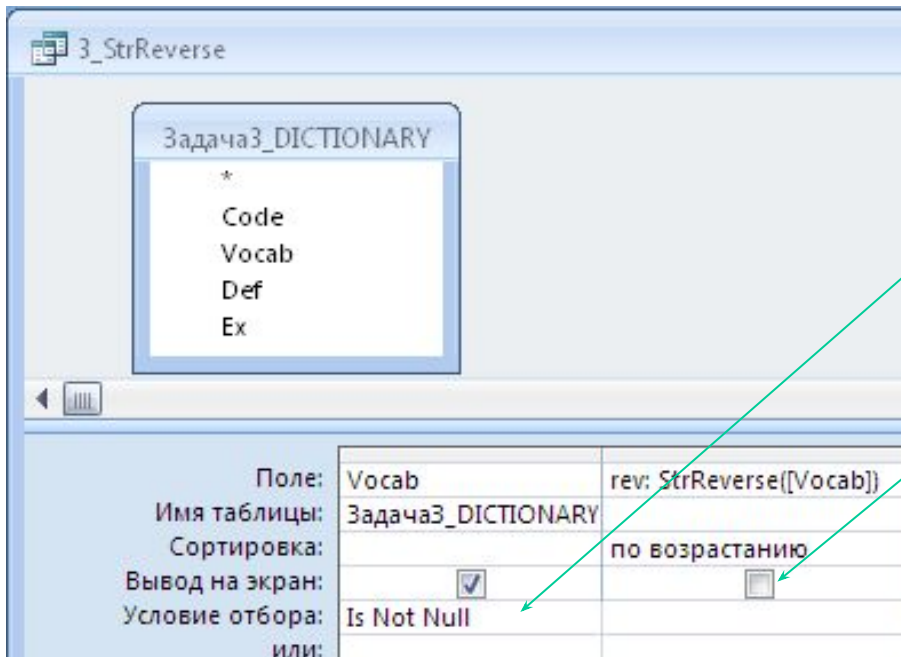
Возвращает строку, содержащую все знаки указанной строки в обратном порядке.

Синтаксис: StrReverse(выражение)

Обязательный аргумент **выражение** — строка, порядок знаков в которой требуется изменить на обратный.

Если **выражение** представляет собой строку нулевой длины (""), возвращается строка нулевой длины. Если аргумент **выражение** имеет значение **Null**, возникает ошибка.

Поскольку возникает синтаксическая ошибка, если функция StrReverse получает как аргумент пустую строку, исключение пустых строк (условие **Is Not Null**) обязательно.



Реверсированная строка не показывается в выдаче.

Задача #4

ЗАДАЧА № 4. Дано: поле SingPlur таблицы **Задача4_PARADYGM_gen** (тип данных: текстовый; каждая запись поля содержит две формы: форму единственного и форму множественного числа родительного падежа некоторой лексики, формы разделены знаком «//»).

Задание: записать содержимое полей Code, LEX, SingPlur в таблицу SEP_Paradygm, распределив содержимое поля SingPlur по двум полям: для каждой записи в поля Singularis и Pluralis записать формы единственного и множественного числа соответственно.

Подсказка: используйте функции **Left**, **Mid**, **InStr**. Тип запроса: **создание таблицы**.

Справку по функциям см. на следующем слайде. ! Обратите внимание: необязательные аргументы НЕ ИСПОЛЬЗУЮТСЯ В РЕШЕНИИ.

Окно запроса:

Pluralis: Mid([SingPlur];InStr([SingPlur];"/")+2)

Singularis: Left([SingPlur];InStr([SingPlur];"/")-1)

4_LeftMidInstr

Задача4_PARADYGM_gen

- * Code
- LEX
- SingPlur

Поле:	Code	LEX	Singularis: Left([SingPlur];InStr([SingPlur];"/")-1)	Pluralis: Mid([SingPlur];InStr([SingPlur];"/")+2)
Имя таблицы:	Задача4_P/	Задача4_P.		
Сортировка:				
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:				

Функция Left

Возвращает подстроку, содержащую определенное аргументом **Длина** количество знаков с левого края указанной строки.

Синтаксис: Left(Строка, Длина)

Функция **Left** имеет следующие аргументы:

- аргумент **Строка**: обязательный аргумент; строковое выражение, с левого края которого будут возвращены знаки; если аргумент **Строка** содержит Null, возвращается значение Null;
- аргумент **Длина**: обязательный аргумент; числовое выражение, указывающее, сколько знаков будет возвращено; если значение этого аргумента равно 0, возвращается пустая строка (""); если значение данного аргумента больше или равно количеству знаков в аргументе **Строка**, возвращается вся строка.

Функция Mid

Возвращает подстроку, которая содержит указанное аргументом **Начало** (и, факультативно – аргументом **Длина**) количество знаков строки.

Синтаксис: Mid(Строка, Начало [, Длина])

Функция **Mid** имеет следующие аргументы:

- аргумент **Строка**: обязательный аргумент; строковое выражение, с левого края которого будут возвращены знаки; если аргумент **Строка** содержит Null, возвращается значение Null;
- аргумент **Начало**: обязательный аргумент; позиция, с которой будет начинаться извлечение знаков в строке, указанной аргументом **Строка**; если значение аргумента **Начало** превышает число знаков в строке, заданной аргументом **Строка**, функция **Mid** возвращает пустую строку ("");
- аргумент **Длина**: необязательный аргумент типа **Variant (Long)**; задает количество возвращаемых знаков; если этот аргумент опущен или количество знаков в строке меньше значения аргумента **Длина** (включая знак, определенный аргументом **Начало**), возвращаются все знаки с позиции, заданной аргументом **Начало**, до конца строки.

Функция InStr

Возвращает номер позиции *первого* вхождения подстроки в строку.

Синтаксис

InStr([*Начальная_позиция*,] *Строка*, *Подстрока* [, *Тип_сравнения*])

Функция **InStr** имеет следующие аргументы:

- аргумент **Начальная_позиция**: необязательный; числовое выражение, которое задает начальную позицию для поиска; если этот аргумент опущен, поиск начинается с первого знака; если аргумент **Начальная_позиция** содержит значение Null, возникает ошибка; если задан аргумент **Тип_сравнения**, аргумент **Начальная_позиция** является обязательным.;
- аргумент **Строка**: обязательный; строковое выражение, по которому выполняется поиск;
- аргумент **Подстрока**: обязательный; искомое строковое выражение;
- аргумент **Тип_сравнения**: необязательный; определяет тип сравнения строк; если значением аргумента **Тип_сравнения** является Null, возникает ошибка; если аргумент **Тип_сравнения** опущен, тип сравнения определяется параметром **Значение**:

Константа	Значение	Описание
vbUseCompareOption	-1	Выполняется сравнение с помощью параметра инструкции Option Compare .
vbBinaryCompare	0	Выполняется двоичное сравнение.
vbTextCompare	1	Выполняется текстовое сравнение.
vbDatabaseCompare	2	Только в Microsoft Office Access 2007. Выполняется сравнение на основе сведений из базы данных.

*Задача #5

ЗАДАЧА № 5. Дано: поле LEX таблицы **Задача5_PREFIXOID** (тип данных: текстовый; каждая запись поля содержит начальные формы слов, в том числе и слов, начинающихся на подстроку **авто...** (слово **авто** тоже относим к этой группе)).

Задание: сделать выборку, содержащую записи поля **LEX** со словами, которые начинаются на подстроку **авто...** и вычисляемое по некоторой функции поле **сочетается_с_АВТО**, содержащее части слова **без подстроки авто**; выборку упорядочить по полю **сочетается_с_АВТО**.

ФРАГМЕНТ ВЫДАЧИ ПО УСЛОВИЮ:

LEX	сочетается_с_АВТО
авто	
автоаматор	аматор
автобаза	база
автобан	бан
автобензоцистерна	бензоцистерна
автобіограф	біограф
автобіографія	біографія
автоброньовик	броньовик
автобудівник	будівник
автобус	бус
автобусик	бусик
автовакцина	вакцина
автовантажник	вантажник
автовідповідач	відповідач
автовізок	візок
автовласник	власник
автоводій	водій
автоводоцистерна	водоцистерна
автовокзал	вокзал

***Задача #6**

ЗАДАЧА № 6. Дано: поле LEX таблицы **Задача6_PREFIXOID** (тип данных: текстовый; каждая запись поля содержит начальные формы слов, в том числе и слов, начинающихся на подстроку **авто** ... (слово **авто** тоже относим к этой группе)).

Задание: создать запрос на обновление записей поля LEX, начинающихся на подстроку **авто...**, так, чтобы часть **АВТО** была написана в верхнем регистре.

ФРАГМЕНТ ВЫДАЧИ ПО УСЛОВИЮ:

Code	LEX
205	австралієць
206	австралійка
207	австралопітек
208	австрієць
209	австрійка
210	австріяк
211	австріячка
212	АВТО
213	АВТОаматор
214	АВТОбаза
215	АВТОбан
216	АВТОбензоцистерна
217	АВТОбіограф
218	АВТОбіографія
219	АВТОброньовик
220	АВТОбудівник
221	АВТОбус
222	АВТОбусик
223	АВТОвакцина
224	АВТОвантажник
225	АВТОвідповідач
226	АВТОвізок

*Задача #7

ЗАДАЧА № 7.

К любой из таблиц заданий 1-6 придумайте свой лингвистический запрос с использованием изученных функций.

Дополнительные материалы: см. папку [дополнительные.материалы\](#)