?????????? ?? MSSQL

Основные команды для администрирования и работы с базами данных MS SQL Server.

???????? ? ???????? ??? ??????

???????? ?? (`CREATE DATABASE`)

Самый простой способ создать новую базу данных. MS SQL Server использует настройки по умолчанию для расположения файлов.

```
CREATE DATABASE MyNewDatabase;
```

Ситуация из жизни: Вы разворачиваете новое веб-приложение, и ему требуется собственная изолированная база данных для хранения пользовательских данных, настроек и контента.

Более продвинутый способ — явно указать пути для файлов данных (.mdf) и логов (.ldf). Это улучшает производительность, если разнести файлы на разные физические диски.

```
CREATE DATABASE MyWebAppDB
ON
( NAME = 'MyWebApp_Data', FILENAME = 'C:\SQLData\MyWebApp.mdf' )
LOG ON
( NAME = 'MyWebApp_Log', FILENAME = 'C:\SQLLogs\MyWebApp.ldf' );
```

???????? ?? (`DROP DATABASE`)

Команда для полного удаления базы данных. **Внимание! Эта команда необратима. Все данные и файлы журнала будут удалены навсегда. Всегда делайте резервную копию перед удалением.**

```
DROP DATABASE MyNewDatabase;
```

Ситуация из жизни: Вы выводите из эксплуатации старое приложение, и его база данных больше не нужна. Удаление освобождает дисковое пространство и системные ресурсы.

???????? ???????????? ? ?????? (`DBCC CHECKDB`)

??????????? ??? ???????????

"Безопасный" режим. Он только сообщает об ошибках, но не пытается ничего исправлять. Это первый шаг в диагностике.

```
DBCC CHECKDB ('YourDatabaseName') WITH NO_INFOMSGS, ALL_ERRORMSGS;
```

Ситуация из жизни: Вы проводите еженедельную плановую проверку баз данных, чтобы убедиться в их "здоровье" без прерывания работы.

?????? ? ??????????? ?????? (`REPAIR_REBUILD`)

Перед ремонтом базу необходимо перевести в однопользовательский режим. Эта команда исправляет незначительные ошибки (например, в индексах) без риска потери данных.

```
-- Перевод в однопользовательский режим
ALTER DATABASE YourDatabaseName SET SINGLE_USER WITH ROLLBACK IMMEDIATE;

-- Выполнение ремонта
DBCC CHECKDB ('YourDatabaseName', REPAIR_REBUILD);

-- Возврат в многопользовательский режим
ALTER DATABASE YourDatabaseName SET MULTI_USER;
```

Ситуация из жизни: `DBCC CHECKDB` нашел ошибки в некластеризованном индексе. `REPAIR_REBUILD` перестроит этот индекс, не затронув сами данные.

?????? ? ????????? ??????? ?????? (`REPAIR_ALLOW_DATA_LOSS`)

Внимание! Это крайняя мера. Используйте ее, только если нет свежей резервной копии. Эта опция может удалить поврежденные страницы данных, чтобы вернуть базу в рабочее состояние. **Потеря данных почти гарантирована.**

```
-- Перевод в однопользовательский режим
ALTER DATABASE YourDatabaseName SET SINGLE_USER WITH ROLLBACK IMMEDIATE;

-- Выполнение ремонта с риском потери данных
DBCC CHECKDB ('YourDatabaseName', REPAIR_ALLOW_DATA_LOSS);

-- Возврат в многопользовательский режим
ALTER DATABASE YourDatabaseName SET MULTI_USER;
```

Ситуация из жизни: Произошел сбой диска, резервной копии нет. База данных не запускается. Чтобы спасти хоть какие-то данные и запустить базу, вы вынуждены применить этот метод, смирившись с потерей части информации.

????? ?????? ? ????????

????? ?????? ?? ?????

Используйте системное представление `INFORMATION_SCHEMA.TABLES` для поиска таблиц по шаблону.

```
SELECT TABLE_SCHEMA, TABLE_NAME
FROM INFORMATION_SCHEMA.TABLES
WHERE TABLE_NAME LIKE '%Users%';
```

Ситуация из жизни: Вы работаете с незнакомой базой данных и вам нужно найти все таблицы, связанные с пользователями. Поиск по части имени (`%Users%`) помогает быстро сориентироваться.

????? ???????? ?? ???? ????????? ?????

Этот скрипт использует курсор и динамический SQL, чтобы найти заданное значение во всех полях типа `(n)char`, `(n)varchar`, `(n)text` во всех таблицах базы.

Измените значение переменной `@SearchStr` на искомое. `%` — это символ-джокер.

```
-- ============================================
-- Автор:      stackoverflow.com (адаптировано)
-- Описание:   Поиск текстового значения во всех текстовых полях всех таблиц БД
-- ============================================

-- Параметр: искомое значение
DECLARE @SearchStr nvarchar(100) = '%ПоискЗначения%';

-- Создаем временную таблицу для хранения результатов
CREATE TABLE #Results (ColumnName nvarchar(370), ColumnValue nvarchar(3630));

SET NOCOUNT ON;

DECLARE @TableName nvarchar(256), @ColumnName nvarchar(128), @SearchStr2 nvarchar(110);
SET  @TableName = '';
-- Заменяем одинарные кавычки для использования в запросе
```

```
SET @SearchStr2 = REPLACE(@SearchStr, '''', '''''');

-- Курсор для перебора всех текстовых полей во всех таблицах
WHILE @TableName IS NOT NULL
BEGIN
    SET @ColumnName = '';
    SET @TableName =
    (
        SELECT MIN(QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME))
        FROM     INFORMATION_SCHEMA.TABLES
        WHERE         TABLE_TYPE = 'BASE TABLE'
            AND    QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME) > @TableName
            AND    OBJECTPROPERTY(
                    OBJECT_ID(
                        QUOTENAME(TABLE_SCHEMA) + '.' + QUOTENAME(TABLE_NAME)
                         ), 'IsMSShipped'
                           ) = 0
    );

    WHILE (@TableName IS NOT NULL) AND (@ColumnName IS NOT NULL)
    BEGIN
        SET @ColumnName =
        (
            SELECT MIN(QUOTENAME(COLUMN_NAME))
            FROM     INFORMATION_SCHEMA.COLUMNS
            WHERE         TABLE_SCHEMA    = PARSENAME(@TableName, 2)
                AND    TABLE_NAME    = PARSENAME(@TableName, 1)
                AND    DATA_TYPE IN ('char', 'varchar', 'nchar', 'nvarchar', 'text', 'ntext')
                AND    QUOTENAME(COLUMN_NAME) > @ColumnName
        );

        -- Выполняем динамический SQL для поиска в текущем поле
        IF @ColumnName IS NOT NULL
        BEGIN
            INSERT INTO #Results
            EXEC
            (
                'SELECT ''' + @TableName + '.' + @ColumnName + ''', LEFT(' + @ColumnName + ',
3630)
                FROM ' + @TableName + ' (NOLOCK) ' +
```

```
                ' WHERE ' + @ColumnName + ' LIKE ''' + @SearchStr2 + ''''
            );
        END
    END
END

-- Выводим результаты
SELECT ColumnName, ColumnValue FROM #Results;

-- Удаляем временную таблицу
DROP TABLE #Results;
```

Ситуация из жизни: К вам пришла жалоба от пользователя 'Иванов Иван' на некорректное отображение данных. Вы не знаете, в какой из сотен таблиц хранится его профиль. Этот скрипт пройдется по всей базе и найдет все таблицы и столбцы, где встречается строка 'Иванов Иван'.

????????? ??????

?????????? ??????? (`UPDATE`)

Команда для изменения данных в существующих строках. **Крайне важно всегда использовать условие `WHERE`, чтобы не изменить все записи в таблице.**

```
-- Увеличить цену на 10% для всех товаров в категории с ID = 5
UPDATE Products
SET Price = Price * 1.10
WHERE CategoryID = 5;
```

Ситуация из жизни: Вам нужно поднять цены на 10% для всех товаров в категории 'Электроника' (с ID = 5).

?????? ????? ?????? (`REPLACE`)

Функция `REPLACE` позволяет найти и заменить часть строки внутри поля. Она часто используется вместе с `UPDATE`.

```
-- Заменить все вхождения старого домена на новый в текстах статей
UPDATE Articles
SET Content = REPLACE(Content, 'http://old-domain.com', 'https://new-domain.com')
WHERE Content LIKE '%http://old-domain.com%';
```

Ситуация из жизни: Компания переехала на новый домен. Вам нужно массово обновить все старые ссылки в текстах статей в базе данных.

Revision #2
Created 4 October 2025 22:59:28 by Admin
Updated 4 October 2025 23:31:27 by Admin