

## Тип системы базы данных

- **OLTP** – Online Transaction Processing  
(интерактивная обработка транзакций)
- **OLAP** - Online Analytical Processing  
(интерактивная аналитическая обработка)

**model** – на ее основе строится пользовательская база данных

**msdb** – содержится необходимая информация для выполнения заданий под управлением SQL Server Agent (Служба Windows в SQL Server)

**tempdb** – временная база данных время жизни которой, продолжительность сеанса SQL Server

## Создание базы данных

- CREATE DATABASE <database\_name>
- [on
- ([NAME = logical\_name, ]
- FILENAME = physical\_file\_name
- [, FILESIZE = size]
- [, MAXSIZE = maxsize]
- [, FILEGROWTH = growth\_increment]])
- [LOG ON
- ([NAME = logical\_name, ]
- FILENAME = physical\_file\_name
- [, FILESIZE = size]
- [, MAXSIZE = maxsize]
- [, FILEGROWTH = growth\_increment]])
- [COLLATE collation\_name]

## Недопустимые символы для имени базы данных

1. “
2. ‘
3. \*
4. /
5. ?
6. :
7. \
8. <
9. >
10. -

```
USE master  
GO
```

```
CREATE DATABASE AFirstBase  
GO
```

```
CREATE DATABASE [AFirstBase]
ON PRIMARY
( NAME = N'AFirstBase', FILENAME =
N'C:\Program Files\ Microsoft SQL
Server\MSSQL14.MSSQLSERVER\
MSSQL\DATA\ AFirstBase.mdf' , SIZE = 2304KB
, MAXSIZE = UNLIMITED, FILEGROWTH =
1024KB )
```

```
LOG ON  
( NAME = N'AFirstBase_log', FILENAME =  
N'C:\Program Files\ Microsoft SQL  
Server\MSSQL14.MSSQLSERVER\MSSQL\DATA  
\AFirstBase_log.LDF' , SIZE = 576KB , MAXSIZE  
= 2048GB , FILEGROWTH = 10%)  
GO
```

```
ALTER DATABASE [AFirstBase] SET  
COMPATIBILITY_LEVEL = 140  
GO
```

Уровень совместимости базы данных



```
IF (1 =  
FULLTEXTSERVICEPROPERTY('IsFullTextInstalle  
d'))  
begin  
EXEC [AFirstBase].[dbo].[sp_fulltext_database]  
@action = 'enable'  
end  
GO
```

Полнотекстовый поиск

```
ALTER DATABASE [AFirstBase] SET  
ANSI_NULL_DEFAULT OFF  
GO
```

Значение по умолчанию для пустого  
столбца NOT NULL

```
ALTER DATABASE [AFirstBase] SET AUTO_SHRINK  
OFF  
GO
```

ON если требуется уменьшается размер базы данных.

Лучше уменьшать вручную DBCC  
SHRINKDATABASE

```
ALTER DATABASE [AFirstBase] SET  
CURSOR_CLOSE_ON_COMMIT OFF  
GO
```

Закрытие курсора при фиксации данных или  
откате при выполнении транзакции

```
ALTER DATABASE [AFirstBase] SET  
CONCAT_NULL_YIELDS_NULL OFF  
GO
```

Если ON то при конкатенации символьных полей получится NULL, если в каком-нибудь столбце имеется значение NULL.

```
ALTER DATABASE [AFirstBase] SET  
NUMERIC_ROUNDABORT OFF  
GO
```

Поведение при потере точности усекается или  
генерирует ошибку - ON

```
ALTER DATABASE [AFirstBase] SET  
QUOTED_IDENTIFIER OFF  
GO
```

Использование идентификаторов в кавычках, а  
не в []

```
ALTER DATABASE [AFirstBase] SET  
RECURSIVE_TRIGGERS OFF  
GO
```

Один триггер не может вызывать другой

```
ALTER DATABASE [AFirstBase] SET  
ENABLE_BROKER  
GO
```

Активация Service Broker



```
ALTER DATABASE [AFirstBase] SET  
AUTO_UPDATE_STATISTICS_ASYNC OFF  
GO
```

Обновление статистики будет запущено, но в фоновом режиме

```
ALTER DATABASE [AFirstBase] SET  
DATE_CORRELATION_OPTIMIZATION OFF  
GO
```

Управляет связью между столбцами типа  
«дата/время»

```
ALTER DATABASE [AFirstBase] SET TRUSTWORTHY  
OFF  
GO
```

Не допустит кода, который может вызвать сбой  
работы сервера, например не допустит кода для  
доступа к внешним данным

```
ALTER DATABASE [AFirstBase] SET  
ALLOW_SNAPSHOT_ISOLATION OFF  
GO
```

Параметр для дублирования информации на  
других серверах

```
ALTER DATABASE [AFirstBase] SET  
PARAMETERIZATION SIMPLE  
GO
```

Как работать с кодом, содержащим параметры

```
ALTER DATABASE [AFirstBase] SET  
READ_COMMITTED_SNAPSHOT OFF  
GO
```

```
ALTER DATABASE [AFirstBase] SET  
HONOR_BROKER_PRIORITY OFF  
GO
```

```
ALTER DATABASE [AFirstBase] SET  
DATE_CORRELATION_OPTIMIZATION OFF  
GO
```

Управляет связью между столбцами типа  
«дата/время»

```
ALTER DATABASE [AFirstBase] SET TRUSTWORTHY  
OFF  
GO
```

Не допустит кода, который может вызвать сбой  
работы сервера, например не допустит кода для  
доступа к внешним данным

ALTER DATABASE [AFirstBase] SET READ\_WRITE  
GO

Параметр для обновления данных

ALTER DATABASE [AFirstBase] SET RECOVERY FULL  
GO

Как будет сервер восстанавливаться после сбоя

```
ALTER DATABASE [AFirstBase] SET MULTI_USER  
GO
```

Доступ к базе (Single\_user), (restricted\_user)

```
ALTER DATABASE [AFirstBase] SET PAGE_VERIFY  
CHECKSUM  
GO
```

При возникновении ошибки ввода/вывода  
предоставляет уведомление

```
ALTER DATABASE [AFirstBase] SET DB_CHAINING  
OFF  
GO
```

Указывает, требуются проверки разрешений  
при ссылке на объекты в другой базе данных



## Создание таблиц

```
CREATE TABLE
  [ database_name . [ schema_name ] . | schema_name . ]
  table_name
  ( { <column_definition> | <computed_column_definition>
    | <column_set_definition> }
    [ <table_constraint> ] [ ,...n ] )
  [ ON { partition_scheme_name ( partition_column_name ) |
filegroup
    | "default" } ]
  [ { TEXTIMAGE_ON { filegroup | "default" } } ]
  [ FILESTREAM_ON { partition_scheme_name | filegroup
    | "default" } ]
  [ WITH ( <table_option> [ ,...n ] ) ]
  [ ; ]
```

## Команда SELECT

```
SELECT [ ALL | DISTINCT ] [ TOP (<выражение>)  
[ PERCENT ] [ WITH TIES ] ]  
<элемент выбора> [, <элемент выбора>]...  
[ INTO <новая таблица> ]  
[ FROM <исходная таблица> [, <исходная  
таблица>]... ]
```

[ WHERE <условие выборки> ]  
[ GROUP BY <условие группировки> ]  
[ HAVING <условие поиска> ]  
[ ORDER BY <список упорядочения> ]  
[ <предложение FOR> ]  
[ OPTION ( <подсказка запроса> [, <подсказка  
запроса>]... ) ]

## Раздел «Сколько хотим»

[ ALL | DISTINCT ] [ TOP (<выражение>)  
[ PERCENT ] [ WITH TIES ] ]

- ALL — разрешает включать в результат запроса дублирующие строки (действует по умолчанию);

- `DISTINCT` — запрещает использовать в результате запроса дублирующие строки;

- **TOP n [PERCENT]**— конструкция, задающая количество строк (**n** — целое число), которые должны быть выведены в запросе. Ключевое слово **PERCENT** означает, что параметр **n** задается в процентах (от всех строк запроса).
- **WITH TIES** – работает вместе с **GROUP BY**. В режиме **WITH TIES** количество возвращаемых строк может превысить значение *expression*.

## Раздел «Чего хотим»

<элемент выбора> [, <элемент выбора>]...

<элемент выбора> ::=

```
{ * | { <имя таблицы> | <имя представления> | <псевдоним таблицы> }. * | {{{ <имя таблицы> | <имя представления> | <псевдоним таблицы> }. } { <имя столбца> | $IDENTITY | $ROWGUID } | <выражение> } [[ AS ] <псевдоним столбца> ] }
```

Символ "\*" означает, что в результат следует включить все поля таблиц, участвующих в запросе.



## Конструкции

{ <имя таблицы> | <имя представления> |  
<псевдоним таблицы> }.\*

используются в том случае, если в запросе участвует несколько таблиц (представлений или псевдонимов таблиц).

*Псевдоним таблицы или представления указывается в предложении FROM.*

## Конструкции

```
{ { { <имя таблицы> | <имя представления> |  
<псевдоним таблицы> }. ] { <имя столбца> |  
$IDENTITY | $ROWGUID } | <выражение> }
```

Позволяют задавать имена столбцов таблиц, создавать столбцы со свойствами IDENTITY и ROWGUIDCOL, выражения - константы, функции, вложенные запросы, т. е. любое допустимое выражение, возвращающее одно значение.

- `<таблица> | <представление> } [[ AS ]`  
`<псевдоним> -`

определяется имя таблицы и ее псевдоним. Использование псевдонима оказывается полезным, если данные выбираются из нескольких таблиц;

- <функция> [ [ AS ] <псевдоним>  
Вместо таблицы или представления может быть указана функция с ее псевдонимом. Функции должны возвращать объект, на который можно ссылаться как на таблицу
- WITH (<подсказка таблицы (hint) > - указывается тип блокировки;

## Раздел **INTO** «Куда хотим»

Данный раздел предназначен для создания таблиц на основе заданного запроса. После слова **INTO** следует указать имя создаваемой таблицы. Если имя таблицы начинается с символа "#", то она считается локальной временной таблицей, если с "##", — то она становится глобальной временной таблицей.

## Раздел **WHERE** «Точнее»

Данный раздел используется для задания критерия отбора строк. В условие могут входить как поля одной или нескольких таблиц, так и данные, к ним не относящиеся. Строится логическое высказывание (предикат).

*Предикат в SQL* — это логическое выражение, которое для каждой исходной строки возвращает значение "истина", "ложь" или значение NULL.

## *Логические операторы:*

- **AND** - конъюнкция;
- **OR** - дизъюнкция;
- **NOT** – логическое отрицание;
- **XOR** – исключающее ИЛИ.



## **Операторы сравнения :**

- "=" — равно;
- "!=" или "<>" — не равно;
- ">" — больше;
- "<" — меньше;
- ">=" — больше или равно;
- "<=" — меньше или равно;
- "!<" — не менее, чем;
- "!">" — не более, чем.

## *Специализированные логические операторы:*

- **ALL** – применяется для сравнения скалярного выражения с одним из столбцов всего набора строк запроса.
- **SOME (ANY)** – применение аналогично предыдущему.

<выражение><операция сравнения> ALL  
(SOME, ANY) <подзапрос>

- **BETWEEN** – позволяет определить, лежит ли данная величина в указанном диапазоне.
- **EXISTS** – оператор возвращает значение "TRUE" если результат указанного в скобках подзапроса содержит хотя бы одну строку.
- **IN** – проверяет, содержится ли заданное выражение в списке, заданном в операнде.

- **EXISTS** – оператор возвращает значение “TRUE” если результат указанного в скобках запроса содержит хотя бы одну строку.
- **IN** – проверяет, содержится ли заданное выражение в списке, заданном в операнде (NOT IN).
- **IS NULL** - проверяет, является ли выражение неопределенным значением (IS NOT NULL).

В случаях поиска записей, для которого не подходят логические операции используются расширенные методы, например инструкция LIKE (NOT LIKE).

% - помещается в начало или конец подстроки;  
- - в строке ищется символ до или после  
подчеркивания

[ ] – позволяет указать ряд значений или  
диапазон

[^ ...] – аналогичен предыдущему, но  
возвращает элементы не попавшие в диапазон.

"#" - используется совместно с ключевым  
словом *ESCAPE* для включения в шаблон  
специальных символов.

Например, чтобы задать шаблон "20%", следует  
записать выражение '20#%' *ESCAPE* '#'

Раздел **GROUP BY** «Разложим по кучкам»  
Позволяет осуществить группировку строк по  
заданному критерию.  
Формат данного раздела имеет следующий вид:

```
[GROUP BY [ALL] <выражение группировки>  
[ ,...n ] [WITH {CUBE | ROLLUP}]]
```

К группам строк можно применять агрегирующие  
функции.

- <выражение группировки> представляет собой условие группировки. Обычно это имя столбца, по которому осуществляется группировка;



- <выражение группировки> представляет собой условие группировки. Обычно это имя столбца, по которому осуществляется группировка;

- ALL — ключевое слово, используемое для вывода списка всех групп, даже если они не попадают под заданное условие, определяемое посредством **WHERE**.

*Если в запросе используются агрегирующие функции, то для таких строк они не выполняются;*

## Агрегирующие функции

- **AVG([ALL | DISTINCT] expression)** - вычисляет среднее значение поля в наборе строк. Игнорируются значения *NULL*. Опция *ALL* означает, что в вычислении участвуют все строки набора, а опция *DISTINCT* — включаются только уникальные строки.
- **COUNT([ALL | DISTINCT] expression)** - осуществляет подсчет количества строк.

- SUM([ALL | DISTINCT] expression)
- MAX([ALL | DISTINCT] expression)
- MIN([ALL | DISTINCT] expression)
- STDEV( expression) - стандартная дисперсия по выборке указанного выражения.

## Раздел **HAVING** «Уточним»

Опция применяется для указания условий группировки, в которых должны использоваться либо поля, указанные также в разделе **GROUP BY**, либо агрегирующие функции.

При этом можно указывать и столбцы, не участвующие в запросе. Ключ **ASC** указывает, что данные сортируются по возрастанию (определен по умолчанию), а **DESC** — данные сортируются по убыванию.

## Раздел COMPUTE «Посчитаем»

Данный раздел позволяет выводить отдельной строкой результаты выполнения агрегирующих функций. Структура раздела имеет следующий вид:

```
COMPUTE{AVG | COUNT | MAX | MIN | STDEV |  
SUM}(expression) } [...n] [BY expression [...n]]
```

## Раздел OPTION «Я круче»

Данный раздел используется редко. Он дает возможность задавать дополнительные параметры, которые оказывают влияние на ход выполнения запроса и его оптимизацию. С помощью OPTION можно также управлять блокировками, устанавливаемыми во время выполнения запроса.



- **RIGHT** [OUTER] - критерий аналогичен предыдущему, только для правой таблицы;
- **FULL** [OUTER] - в запрос будут включены все строки правой и левой таблиц. Данная операция соответствует теоретико-множественному объединению;
- **ON** <условие> - задает логическое условие связывания двух таблиц;

CROSS JOIN <таблица>

Декартово произведение таблиц и условие ON не используется.

{ CROSS | OUTER } APPLY <таблица>

Вариант APPLY.

Если задано ключевое слово CROSS, то будут возвращены строки, для которых найдено соответствие в правой таблице, если - OUTER, то возвращаются строки, для которых соответствие не найдено.