Язык реляционной базы данных в **SQL Server** называется **Transact-SQL** (**T-SQL**).

Операторы SQL делятся на несколько непересекающихся групп: **DDL** (Data Definition Language, язык определения данных), **DCL** (Data Control Language, язык управления данными), **DML** (Data Manipulation Language, язык манипулирования данными) и **TCL** (Transaction Control Language, язык управления транзакциями)

Операторы DDL предназначены для создания, удаления и изменения объектов БД или сервера СУБД. В этой книге будут в основном рассматриваться операторы уровня БД. DDL включает три оператора: **CREATE**, **ALTER**, **DROP** (рис. 3.2).



Рис. 3.2. Операторы DDL SQL

Оператор **CREATE** предназначен для создания объектов БД или сервера СУБД. Структура оператора представлена на рис. 3.3.



Рис. 3.3. Структура оператора CREATE

Каждый объект БД имеет тип (например, процедура, таблица, представление и т. д.), имя, а также ряд дополнительных параметров (на рис. 3.3 обозначены словом «дополнение»), характеризующих данный тип объекта. В некоторых случаях дополнительные параметры могут не указываться – в этом случае для объекта автоматически устанавливаются значения параметров объекта по умолчанию.

На рис. 3.4 представлен пример создания БД (тип **database**) с именем **BSTU** – объекта сервера СУБД. БД **BSTU** располагается в двух файлах операционной системы с именами, указанными параметрами **filename,** имеющими логические имена (параметр с именем **name**) соответственно **BSTU** и **BSTU\_log**. Параметры **size**, **maxsize** и **filegrowth** указывают на начальный размер, максимальный размер, а также приращение размера файла.

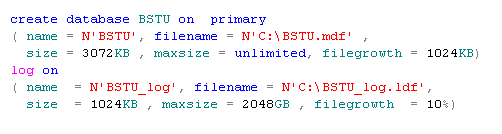
****

Рис. 3.4. Пример использования оператора CREATE

для создания базы данных

На рис. 3.5 приведен пример создания таблицы (тип **table**) – объекта БД. Таблица имеет имя **AUDITORIUM** и состоит из пяти столбцов. Для столбца с именем **AUDITORIUM** указывается ограничение целостности первичный ключ (primary key) с именем **AUDITORIUM\_PK**.

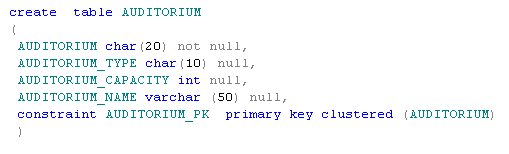
****

Рис. 3.5. Пример использования оператора CREATE

для создания таблицы

Для модификации существующих объектов БД или сервера СУБД применяется оператор **ALTER**. Структура оператора представлена на рис. 3.7.



Рис. 3.7. Структура оператора ALTER

Удалить (иногда применяют слова «уничтожить», «разрушить») существующий объект сервера или БД можно с помощью оператора DROP. Структура оператора представлена на рис. 3.11.



Рис. 3.11. Структура оператора DROP

Пример на рис. 3.12 демонстрирует двукратное применение оператора **DROP**.

****

Рис. 3.12. Пример использования операторов DROP  
 для удаления таблицы и функции

В первом случае удаляется таблица (тип **table**) с именем **TRACHER**. Во втором удаляется функция (**function**) с именем **CALC\_CAPACITY**.

Операторы DML предназначены для работы с одним (наиболее важным) типом объектов БД – таблицами. DML включает четыре оператора: **SELECT**, **INSERT**, **DELETE**, **UPDATE** (рис. 3.13). Иногда к этой группе относят оператор **TRUNCATE**.



Рис. 3.13. Операторы DML SQL

Наиболее мощным DML-оператором является **SELECT**. Он позволяет выбрать множество строк из одной или нескольких таблиц. При успешном выполнении этого оператора формируется результирующий набор, представляющий собой множество однотипных (с одинаковыми столбцами) строк. В общем случае результирующий набор может содержать ни одной, одну или более строк. Структура оператора SELECT достаточно сложная и детально будет разобрана позже, но в первом приближении ее можно представить как на рис. 3.14.



Рис. 3.14. Структура оператора SELECT

Любой оператор SELECT содержит список (на рис. 3.14 обозначен словом «список»), определяющий перечень столбцов (в общем случае и содержимое) результирующего набора. Дополнительная часть оператора (на рис. 3.14 обозначена словом «дополнение») описывает множество строк из одной или нескольких таблиц, служащее источником для формирования результирующего набора.

На рис. 3.15 приведены примеры использования оператора SELECT.

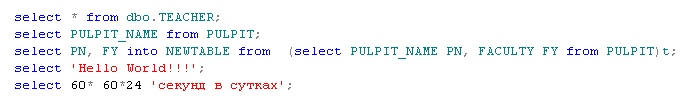
****

Рис. 3.15. Примеры использования операторов SELECT

В первом случае в результате выполнения оператора SELECT будет сформирован результирующий набор, состоящий из всех строк таблицы **dbo.TEACHER**, причем строки будут содержать все столбцы этой таблицы.

Второй оператор SELECT сформирует результирующий набор, состоящий из одного столбца с именем **PULPIT\_NAME** таблицы **PULPIT**.

Третий оператор SELECT демонстрирует создание новой таблицы с именем **NEWTABLE**, включающей два столбца с именами **PN** и **FY**. Cодержимое новой таблицы будет определяться результирующим набором, сформированным в результате выполнения другого (внутреннего) оператора **SELECT**, записанного в скобках после ключевого слова **FROM**.

Четвертый и пятый операторы формируют результирующие наборы, состоящие из одной строки, содержимое которой определяется значением выражений, указанных после ключевого слова **SELECT**.

Добавить одну или несколько строк в существующую таблицу можно с помощью оператора **INSERT**. Структура этого оператора представлена на рис. 3.16.



Рис. 3.16. Структура оператора INSERT

Ключевое слова **INTO** указывает на то, что далее следует имя таблицы, в которую будут добавляться строки.

На рис. 3.17 приведены четыре примера использования оператора INSERT.

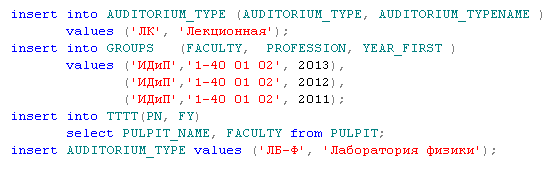
****

Рис. 3.17. Примеры использования операторов INSERT

В первом примере оператор **INSERT** добавляет одну строку в таблицу с именем **AUDITORIUM\_TYPE.** Добавляемая строка содержит столбцы с именами **AUDITORIUM\_TYPE** и **AUDITORIUM\_TYPENAME,** а соответствующие им значения указаны в скобках после ключевого слова **VALUES**.

Во втором примере добавляется три строки в таблицу **GROUPS**.

Третий пример на рис. 3.17 демонстрирует добавление в таблицу **GROUPS** строки из результирующего набора, сформированного оператором SELECT.

Последний пример на рис. 3.18 добавляет одну строку. При этом предполагается, что порядок в списке значений, указанных в скобках за ключевым словом VALUES, соответствует исходному (указанному в операторе CREATE) порядку столбцов таблицы **AUDITORIUM\_TYPE**.

Для удаления строк из таблицы предназначен оператор DELETE. Структура этого оператора приведена на рис. 3.18.



Рис. 3.18. Структура оператора DELETE

В результате выполнения оператора DELETE из таблицы удаляется ноль, одна или несколько строк.

На рис. 3.19 приведены три примера применения оператора DELETE.

В первом случае удаляются все строки из таблицы с именем **TTTT**.

Во втором примере из таблицы SUBJECT (ключевое слово FROM может быть опущено) удаляются строки, которые в столбце PULPIT имеют значение, указанное в кавычках после символа =.

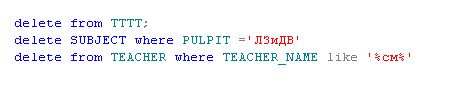
****

Рис. 3.19. Примеры использования операторов DELETE

Третий пример демонстрирует удаление из таблицы **TEACHER** строк, в столбце **TEACHER\_NAME** которых содержится подстрока, обрамленная символами % в выражении, следующем за ключевым словом LIKE.

Для изменения строк таблицы предназначен оператор UPDATE. Структура этого оператора приведена на рис. 3.20.



Рис. 3.20. Структура оператора UPDATE

В результате выполнения оператора UPDATEв таблице изменяется ноль, одна или несколько строк.

На рис. 3.21 приведены два примера применения оператора UPDATE.

****

Рис. 3.21. Примеры использования операторов UPDATE

В первом случае изменяются строки таблицы **AUDITORIUM**. Во всех строках таблицы, значение столбца **AUDITORIUM\_TYPE** которых равно строке **ЛБ-К,** числовое значение в столбце **AUDITORIUM\_CAPACITY** умножается на 1.15.

Во втором примере во всех строках таблицы **TEACHER** значение в столбце **PULPIT** изменяется на строку **ИСиТ**.

БД представляет собой набор файлов операционной системы трех типов: первичный файл (расширение **mdf**), вторичные файлы (**ndf**) и файлы журнала транзакций (**log**). Все файлы БД, кроме файлов журнала транзакций, распределены по файловым группам. Файловые группы – это поименованный набор файлов БД.