Существует 3 группы операторов:

DDl (определение данных)

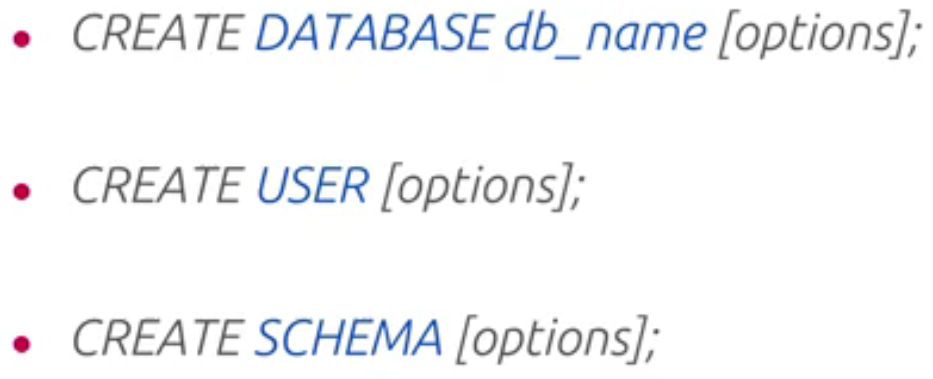
DML (манипулирование данными)

DCL (управление данными)

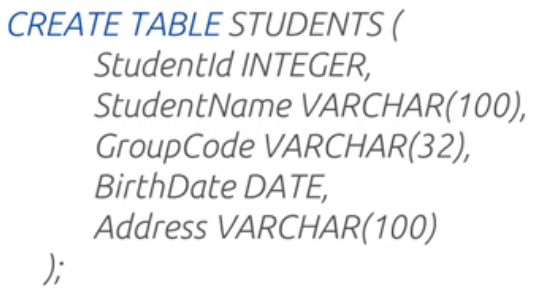
**DDL**

**CREATE**

Создание базы данных

****

Чтобы создать таблицу используем оператор CREATE TABLE

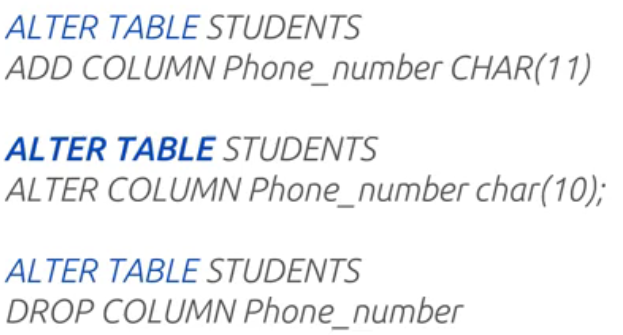
****

**ALTER**

После создание таблицы её структуру можно менять, для этого используем оператор

ALTER TABLE tableName

{ADD|DROP|ALTER} COLUMN

****

**DROP**

DROP TABLE tableName – удаляет всю таблицу

**DML**

**SELECT**

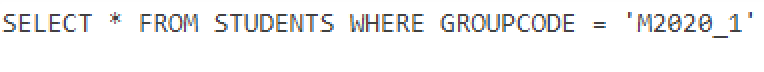
Всего один оператор для любого поиска.

**** выводит всю таблицу

****

если нужна не вся таблица, а конкретные атрибуты

Можно также добавить условия

****

Вот есть у нас таблица со студентами, у каждого из которых есть GROUPCODE, если мы пишем запрос , то выйдут группы всех студентов, соответственно они будут повторяться. Чтобы указать это в явном виде пишем ALL перед именем полей.



Если же мы хотим выводить только уникальные строки, используем ключевое слово DISTINCT



Для сортировки используем оператор ORDERED BY



Сортировать можно по возрастанию ASK и по убыванию DESK, по умолчанию происходит сортировка по возрастанию, чтобы это изменить дописываем в конец DESC



Можно указать несколько критериев, тогда сортировка производится по первому, а в случае, если значения равны, то досортировывается по второму



Для удобства можно переименовывать столбцы только для выборки

****

В таком случае выданная таблица будет иметь столбцы Name и Group

Также можно выводит результат выполнения процедур, а ещё создавать собственные столбцы



т

**WHERE**

Позволяет создавать условия для поиска, например с помощью операций сравнения

(=, <>, >, >=, <, <=)

****

Также можно проверить значение на принадлежность множеству, для этого используем слово IN



Со значением NULL нужно быть аккуратным, к неприменимы операции сравнения (результат всегда неопределён). Чтобы проверить используем NULL или NOT NULLи



Строки можно искать по шаблону, для этого используется ключевое слово **LIKE**

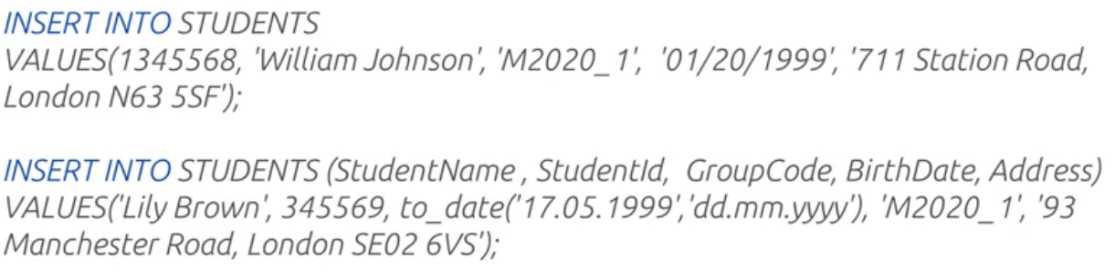
% - любая последовательность любых символов

\_ - один любой символ



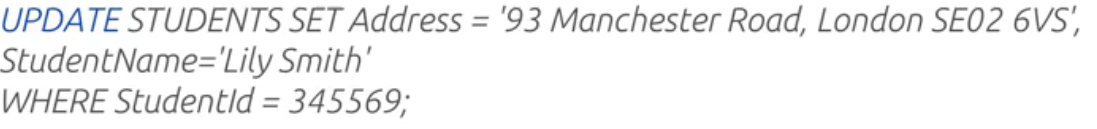
Если условие множественное, то можно использовать логические операторы NOT, AND, OR, а также скобки чтобы скомпоновать их.

**INSERT**

****

**UPDATE**

Изменяет существующие значения в таблице

****

**DELETE**

Удаляет строки из таблицы в соответствии с каким либо условием. Если применить команду без параметров, то она удалит все строки из таблицы. DELETE FROM TABLE T1

****

**TRUNCATE**

Удаляет все записи из таблицы

****

**DСL**

**GRANT**

Дать права (может ли пользователь читать данные из таблицы или исполнить функцию)

**REVOCE**

Отнять права соответственно

**Ограничения целости**

Ограничения пишутся после имени и типа поля в любом порядке без запятых

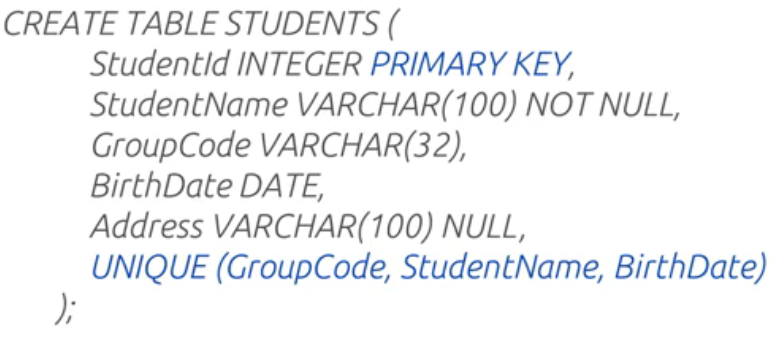
По умолчанию значения полей могут быть не заданы (могут быть NULL), это можно прописать и в явном виде, дописав после типа данных NULL

Если же значение поле должно быть определено, указываем, что поле NOT NULL

Для уникальных значений используем ограничение UNIQUE

Если уникальным должно быть не поле, а группа полей, то прописываем ограничение после их объявления UNIQUE (ATTR\_1, ATTR\_2, ATTR\_3)

Довольно сильное ограничение – PRIMARY KEY, может быть только один в таблице, требует уникальности значений и NOT NULL q, в качестве праймари кей может выступать комбинация из нескольких значений PRIMARY KEY (ATTR\_1, ATTR\_2)



**Ограничения ссылочной целости**

Ссылочная целостность – вот есть у меня таблица со списком групп ST\_GROUP, а есть отдельная таблица со списком студентов, как гарантировать, что в графе группы у студента не будет P3116? Для этого мы используем ограничение ссылочной целостности



Внешний ключ и связываемый столбец должны иметь одинаковый тип данных.

Более того, ссылочное поле должно быть UNIQUE либо PRIMARY KEY.

Если строки родительской таблицы меняются либо удаляются, то автоматически происходит поиск связанных строк в таблице, которая ссылается на родительскую. Дальнейшие действия зависят от настроек BD, например удаление строк в родительской таблице может быть запрещено, или из дочерней таблицы удалят все связанные записи.

**Ограничение допустимого диапазона значений**

Можем заявить, что какое-либо поле должно быть больше 15, или не содержать цифр и т.д.



**ФУНКЦИИ**

**COUNT**

Можно прописать , чтобы определить кол-во записей

****

****

Можно искать по той части таблицы, которая остается после условия

выводит кол-во уникальных значений

**MIN/MAX**





**AVG**

Б

**SUM**

Б

**Агрегирование с группировкой**

В качестве критерия группировки можно указать столбец или несколько столбцов, тогда строки, в которых значения совпадают будут распределены в одну группу (то есть у нас будет несколько групп, в которых определённый критерий совпадает). Для этого используем GROUP BY



Но лучше будет сделать вот так



Но слева помещаем только название столбца, по которому производили группировку

Возможно пропускать не все группы, а только по каким то условиям, используя слово HAVING



В примере не будут пропущены группы, в которых больше 8 элементов

**Вложенный запрос**

Могу также прикрутить к одной таблице столбцы из другой



**Оператор EXIST**

****

Возвращает значение true или false

**Операторы ANY/ALL**

Используются когда есть вложенный запрос, который возвращает один столбец, и его мы с чем то сравниваем

****

Двойка сравнивается с результатом внутреннего запроса