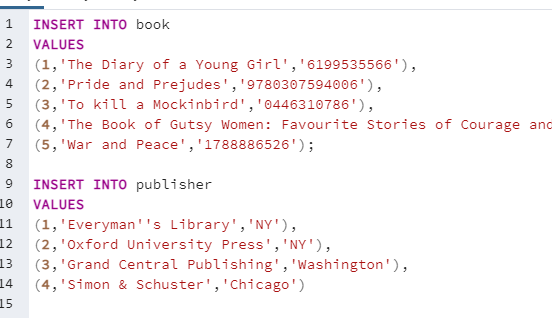
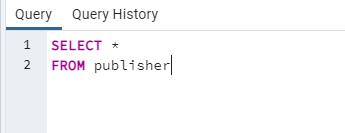
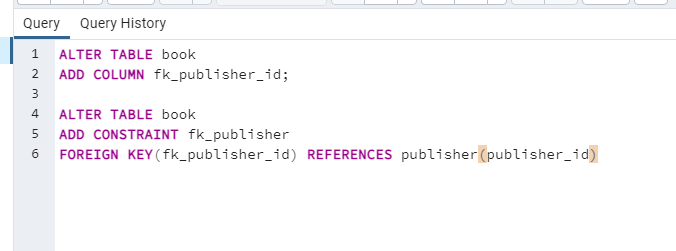
1. Добавление информации в таблицу



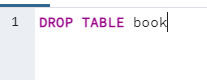
1. Выбрать все из таблицы



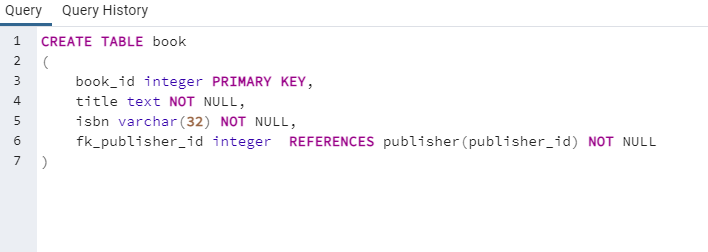
1. Создаем в таблице еще одну колонку, потом создаем ограничение и ссылаемся на другую таблицу



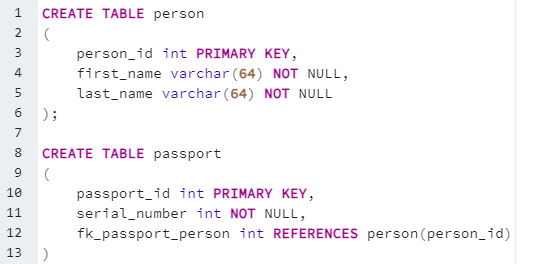
1. Удаляем таблицу



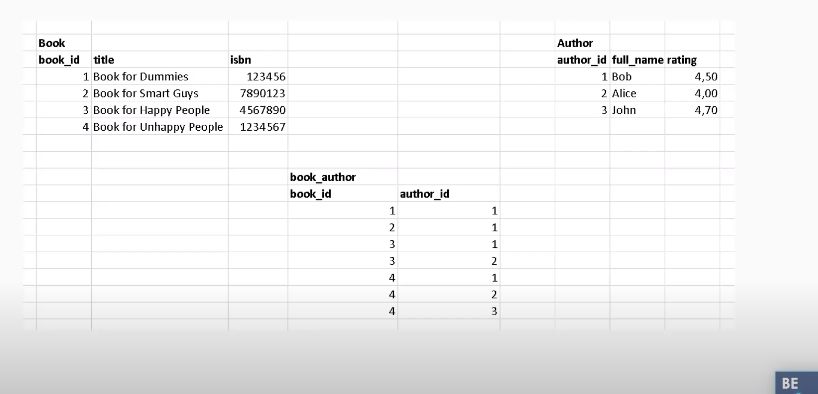
1. Создаем таблицу и связываем ее столбец с другой таблицей(пример один ко многим)

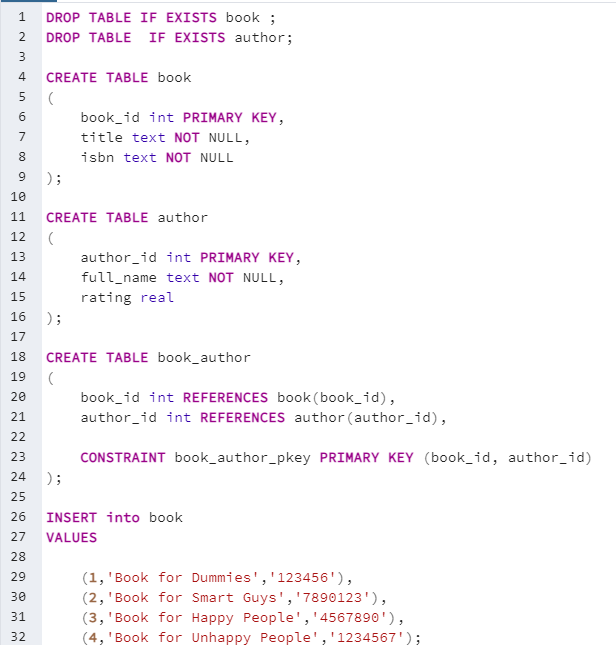


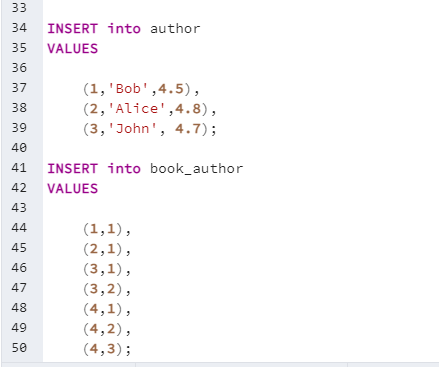
1. Создаем две таблицы и связываем(один к одному)



1. Чтобы создать отношение многие ко многим нужно завести дополнительную таблицу

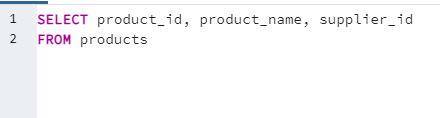




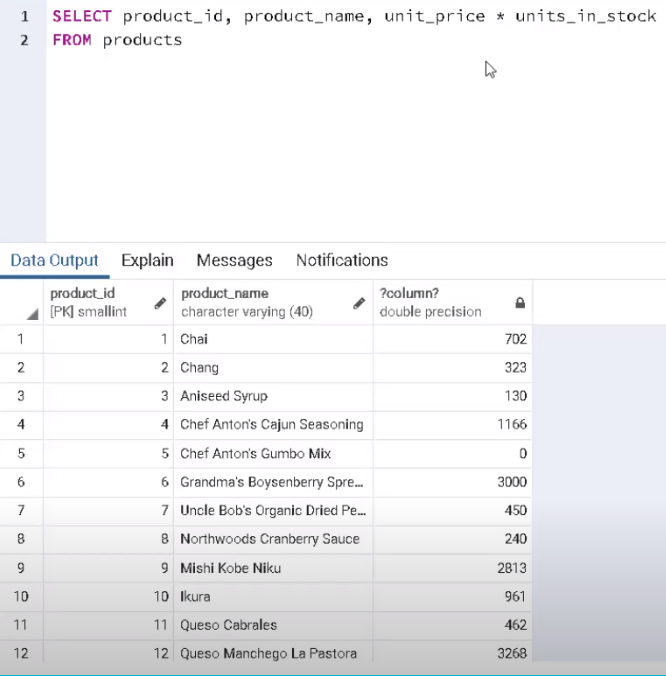


**ПРОСТАЯ ВЫБОРКА ИЗ ОДНОЙ ТАБЛИЦЫ**

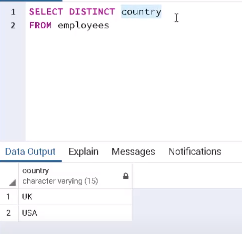
1. Простая выборка



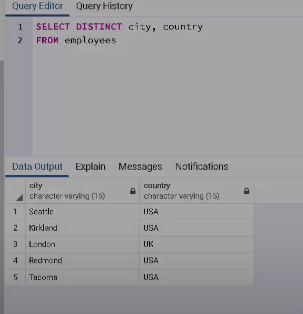
1. Можно использовать арифметические операции /\*-+^



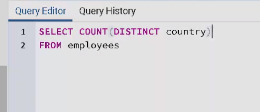
1. Можно выбрать уникальные данные из колонки (DISTINCT)



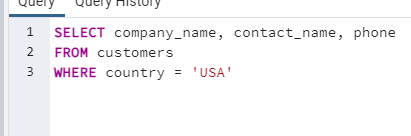
1. Так же можно применять к нескольким столбцам



1. Чтобы подсчитать число чего то, используем COUNT



1. Можно задавать условия с помощью Where

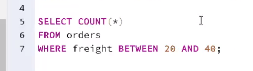


1. В условиях активно используется OR, AND





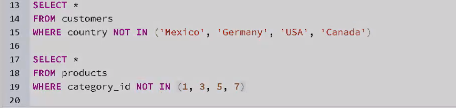
1. Можно использовать BEETWEN ( между двумя значениями, причем значения входят в интервал)



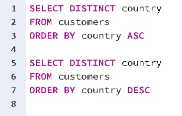
1. Оператор IN



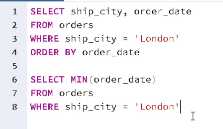
1. Часто используется с NOT IN



1. Для сортировки используется ORDER BY (по умолчанию сортирует по возрастанию ASC, убывание DESC)



1. Для сортировки также используется MIN, MAX, AVG-среднее значение



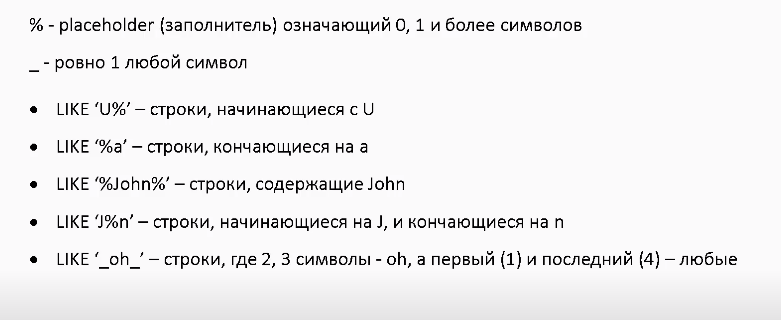


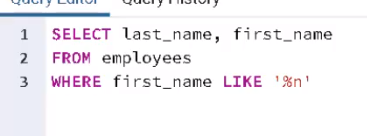


1. Для суммирования можно использовать функцию SUM

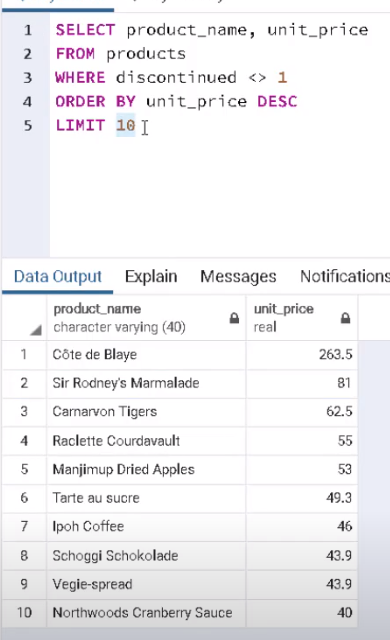


1. Использование LIKE для поиска по шаблону

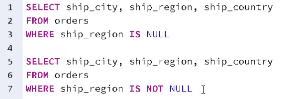




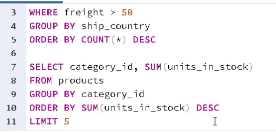
1. Использование LIMIT для вывода определенного числа строк



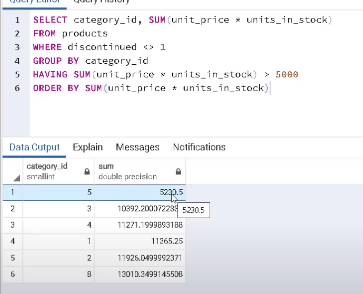
1. Найти все строки которые содержат NULL или наоборот не содержат можно при помощи IS NULL , IS NOT NULL



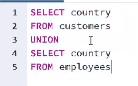
1. Группировать можно при помощи GROUP BY (Представим много людей в комнате и перед тем как посчитать какой средний рост жителей США мы сначала должны сгруппировать этих людей из США и потом уже считать средний рост по этой категории)



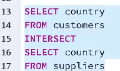
1. Вторичный фильтр HAVING, когда нам необходимо еще раз применить фильтр после WHERE



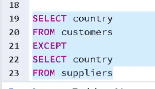
1. Объединение двух селектов UNION ( UNION ALL – если хотим не только уникальные, а повторящиеся)



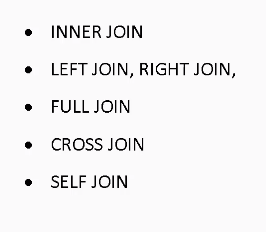
1. Пересечение двух селектов INTERSECT



1. Показывает то, чего нет в другом селекте EXCEPT



**СОЕДИНЕНИЯ JOINS**

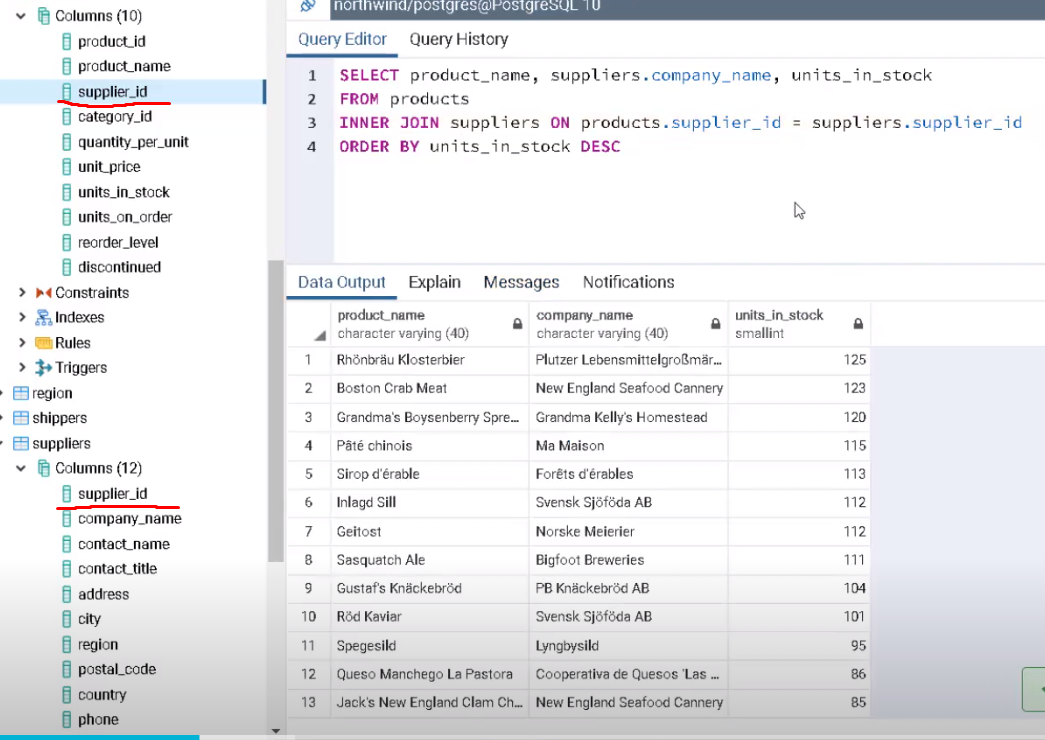


Если не понятно, то лучше посмотреть обобщенный урок 3.1

**В случае одинакового названия столбца в разных таблицах можно обратиться напрямую**

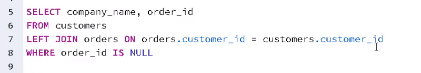


1. Внутреннее соединение INNER JOIN



Покажет только те строки, на которые есть ссылка, остальные опустит

1. Left, right JOIN соединение (если ко всем строкам есть ключи, то результат будет похож на INNER JOIN)



Применяется при поиске строк, которым не присвоен ключ( тип предприятия которые не делали заказы и у них в статусе стоит NULL)



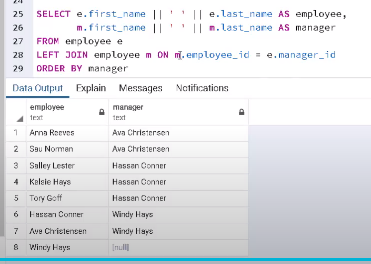
1. Есть еще CROSS JOIN, FULL JOIN

При кросс не нужен ON

1. SELF JOIN ( для построения некой внутренней иерархии)

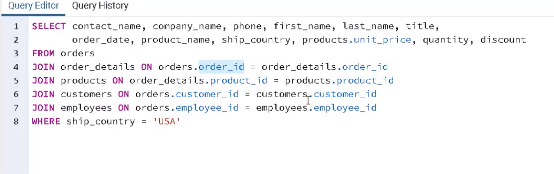


У каждого сотрудника есть менеджер

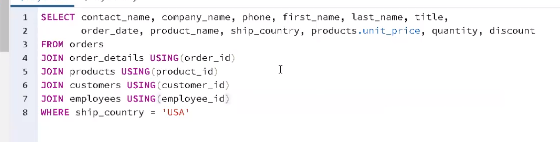


1. USING & NATURAL JOIN ( Для упрощения записи)

было



Стало



Работает только тогда, когда ключи названы одинаково

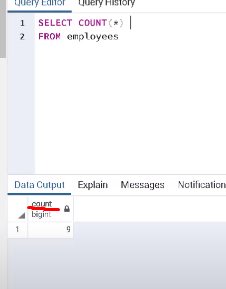
Можно еще короче

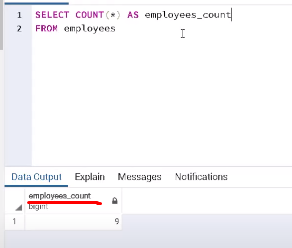


Работает как обычный JOIN, и сравнение происходит по всем столбцам которые названы одинаково

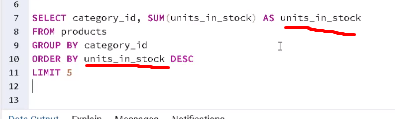
**НИКОГДА НЕ ИСПОЛЬЗОВАТЬ NATURAL JOIN, ТАК КАК ЗАБАГОВАН**

1. AS – псевдоним

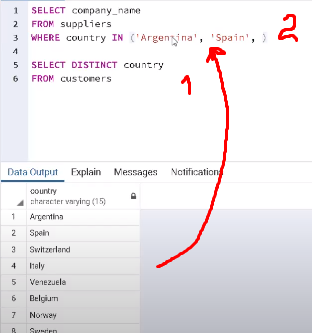
было

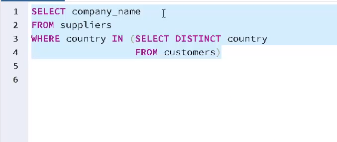
стало

Псевдоним нельзя использовать в WHERE, но можно использовать в ORDER BY и GROUP BY

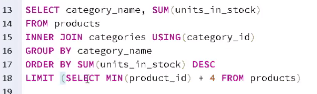


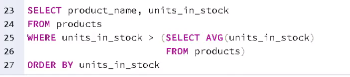
**ПОДЗАПРОСЫ**

1.  Так не рационально, вдруг много запросов, поэтому делаем так:

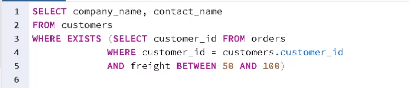


Подзапрос может находиться в разных местах:



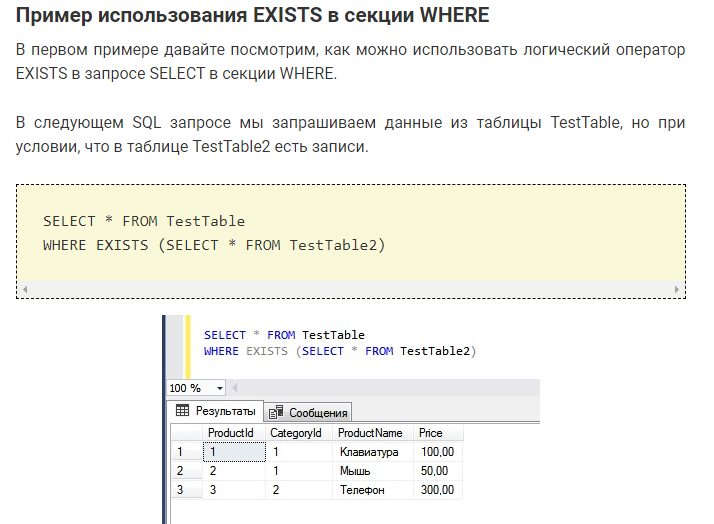


1. WHERE EXISTS

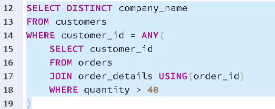


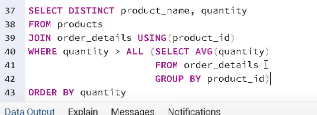
**EXISTS** – это логический оператор языка T-SQL, который принимает и обрабатывает вложенный SQL запрос (SELECT) с целью проверки существования строк. В качестве результата возвращает значения (Boolean):

* TRUE – если вложенный запрос возвращает хотя бы одну строку;
* FALSE – если вложенный запрос не возвращает строк

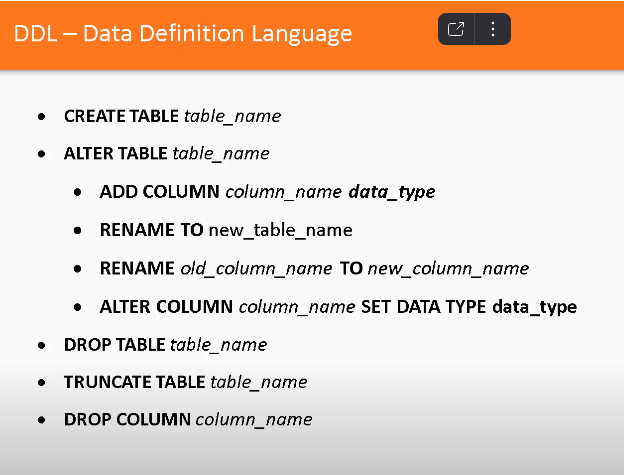


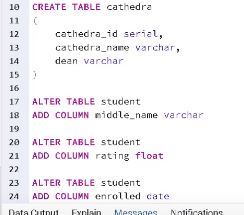
1. Подзапросы с ANY, ALL ( для сравнения с подзапросом)

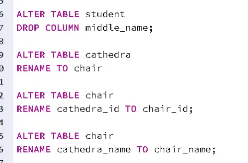


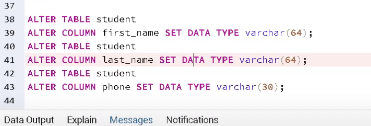


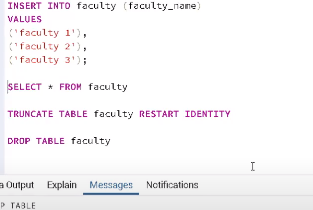
**DDL управление базами данными**



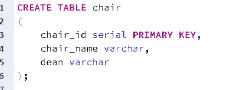




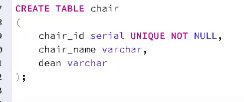




1. Ограничение PRIMARY KEY( дает ограничение уникальности, нельзя использовать NULL)

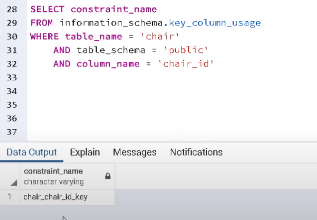


UNIQUE (накладывает ограничение только уникальности, а отдельно прописав NOT NULL тогда и ограничение NULL)



Разница в том, что PRIMARY KEY может быть только один на всю таблицу, а UNIQUE хоть сколько

1. Можно посмотреть какие есть ограничения в нашей таблице:



1. Удалить ограничение можно командой:



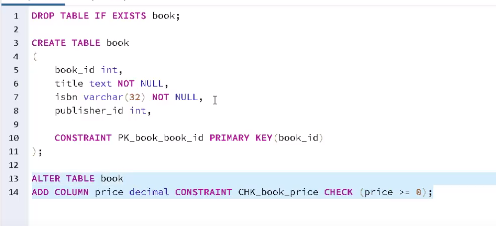
1. Добавить ограничение:



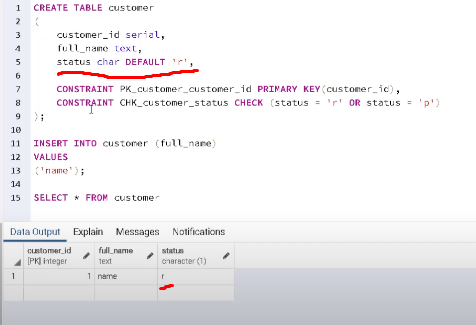
1. FOREIGN KEY (ограничение внешнего ключа)



1. CHECK (ограничение с условием, в данном случае нельзя записать в строку число меньше 0)



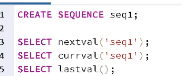
1. DEFAULT ( ограничение, в строчке без записи будет автоматически появляться строка прописанного формата)



Снять ограничение можно:



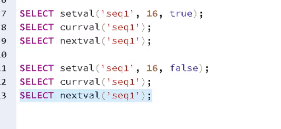
1. Sequences ( Последовательности)



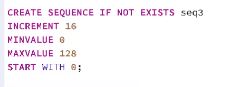
Nextval – генерирует шаг последовательности

Currval – показывает текущее значение

Lastval – показывает последнее сгенерированное в данной сессии



Setval указываем как начинается последовательность, а true или false увеличиваем или уменьшаем значение в следующем шаге



Можно указывать так же инкремент и другие условия

Можно делать рестарт и переименовывать



Так же удалить последовательность



1. Создать последовательность и применить ее к какому нибудь столбцу



И если в этот столбец ничего не вставляют и нам нужно сделать шаг последовательности, то



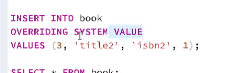
1. **БОЛЕЕ ПРОДВИТЫЙ СПОСОБ ЗАДАНИЯ ПОСЛЕДОВАТЕЛЬНОСТИ**

,



А вместо ALWAYS можно писать DEFAULT

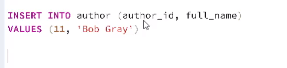
Обойти это ограничение можно с помощью



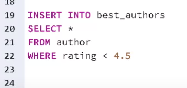
1. Вставить данные в таблицу можно так ( но только если мы хотим вставить данные во все колонки таблицы сразу):

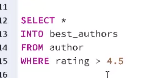


А если нужно вставить в определенные колонки:



1. Вставить данные из одной таблицы в другую можно так:



А если с созданием новой таблицы то так:  


1. Если мы хотим изменить строку в таблице:

(во второй строке мы указываем что хотим там видеть, а в третьей строке указываем на строку где надо изменить)

1. Удалить строки:



А удалить все строки:



1. Показать какие данные были изменены в последней итерации

