1. **Module 02. Simple queries, join tables, subqueries**
   1. **Работа с типами данных Date, NULL значениями, трехзначная логика. Возвращение определенных значений в результатах запроса в зависимости от полученных первоначальных значений результата запроса. Высветка в результатах запроса только определенных колонок.**
      1. Выбрать в таблице **Orders** заказы, которые одновременно удовлетворяют условиям:
         * были доставлены после 6 мая 1998 года (колонка ShippedDate) включительно
         * доставлены с ShipVia >= 2.

Формат указания даты должен быть верным при любых региональных настройках, согласно требованиям статьи “[Writing International Transact-SQL Statements](https://docs.microsoft.com/en-us/sql/relational-databases/collations/write-international-transact-sql-statements)” в Books Online раздел “Accessing and Changing Relational Data Overview”.

Этот метод использовать далее для всех заданий.

Запрос должен высвечивать только колонки:

* + - * OrderID
      * ShippedDate
      * ShipVia.

Пояснить почему сюда не попали заказы с NULL-ом в колонке ShippedDate.

* + 1. Написать запрос, который выводит только недоставленные заказы из таблицы **Orders**.

Запрос должен высвечивать только колонки:

* + - * OrderID
      * ShippedDate

Для колонки ShippedDate вместо значений NULL выводить строку ‘Not Shipped’ – для этого использовать системную функцию CASЕ.

* + 1. Выбрать в таблице **Orders** заказы, которые удовлетворяют хотя бы одному из условий:
       - были доставлены после 6 мая 1998 года (колонка ShippedDate) не включая эту дату
       - еще не доставлены.

Запрос должен высвечивать только колонки:

* + - * OrderID (переименовать в Order Number)
      * ShippedDate (переименовать в Shipped Date)

Для колонки ShippedDate вместо значений NULL выводить строку ‘Not Shipped’, для остальных значений высвечивать дату в формате по умолчанию.

* 1. **Использование операторов IN, DISTINCT, ORDER BY, NOT**
     1. Выбрать из таблицы **Customers** всех заказчиков, проживающих в USA и Canada. Запрос сделать с только помощью оператора IN.

Высвечивать колонки с именем пользователя и названием страны в результатах запроса.

Упорядочить результаты запроса по имени заказчиков и по месту проживания.

* + 1. Выбрать из таблицы **Customers** всех заказчиков, не проживающих в USA и Canada. Запрос сделать с помощью оператора IN.

Высвечивать колонки с именем пользователя и названием страны в результатах запроса.

Упорядочить результаты запроса по имени заказчиков.

* + 1. Выбрать из таблицы **Customers** все страны, в которых проживают заказчики.

Страна должна быть упомянута только один раз.

Cписок стран должен быть отсортирован по убыванию.

Не использовать предложение GROUP BY.

Высвечивать только одну колонку в результатах запроса.

* 1. **Использование оператора BETWEEN, DISTINCT**
     1. Выбрать все заказы (OrderID) из таблицы **Order Details** (заказы не должны повторяться), где встречаются продукты с количеством от 3 до 10 включительно – это колонка Quantity в таблице Order Details.

Использовать оператор BETWEEN.

Запрос должен высвечивать только колонку OrderID.

* + 1. Выбрать всех заказчиков из таблицы **Customers**, у которых название страны начинается на буквы из диапазона b и g.

Использовать оператор BETWEEN.

Проверить, что в результаты запроса попадает Germany.

Запрос должен высвечивать только колонки

* + - * CustomerID
      * Country

Результат запроса должен быть отсортирован по Country.

* + 1. Выбрать всех заказчиков из таблицы **Customers**, у которых название страны начинается на буквы из диапазона b и g.

Не используя оператор BETWEEN.

Запрос должен высвечивать только колонки

* + - * CustomerID
      * Country

Результат запроса должен быть отсортирован по Country.

С помощью опции “Execution Plan” определить какой запрос предпочтительнее 2.3.2 или 2.3.3. Для этого надо ввести в скрипт выполнение текстового Execution Plan-a для двух этих запросов. Результаты выполнения Execution Plan надо ввести в скрипт в виде комментария и по их результатам дать ответ на вопрос – по какому параметру было проведено сравнение.

Запрос должен высвечивать только колонки CustomerID и Country и отсортирован по Country.

* 1. **Использование оператора LIKE, строковых функций**
     1. В таблице **Products** найти все продукты (колонка ProductName), где встречается подстрока 'chocolade'. Известно, что в подстроке 'chocolade' может быть изменена одна буква 'c' в середине - найти все продукты, которые удовлетворяют этому условию.

Подсказка: результаты запроса должны высвечивать 2 строки.

* + 1. Для формирования алфавитного указателя Employees высветить из таблицы **Employees** список только тех букв алфавита, с которых начинаются фамилии Employees (колонка LastName ) из этой таблицы.

Алфавитный список должен быть отсортирован по возрастанию.

* 1. **Join tables, subqueries**
     1. Определить продавцов, которые обслуживают регион 'Western' (таблица **Region**). Результаты запроса должны высвечивать поля:
        + 'LastName' продавца
        + название обслуживаемой территории ('TerritoryDescription' из таблицы **Territories**).

Запрос должен использовать JOIN в предложении FROM.

Для определения связей между таблицами **Employees** и **Territories** надо использовать графические диаграммы для базы Northwind.

* + 1. Высветить в результатах запроса имена всех заказчиков из таблицы **Customers** и суммарное количество их заказов из таблицы **Orders**.

Принять во внимание, что у некоторых заказчиков нет заказов, но они также должны быть выведены в результатах запроса.

Упорядочить результаты запроса по возрастанию количества заказов.

* + 1. Высветить всех поставщиков колонка CompanyName в таблице **Suppliers**, у которых нет хотя бы одного продукта на складе (UnitsInStock в таблице Products равно 0).

Использовать вложенный SELECT для этого запроса с использованием оператора IN.

Можно ли использовать вместо оператора IN оператор '=' ?

* + 1. Высветить всех продавцов, которые имеют более 150 заказов.

Использовать вложенный коррелированный SELECT.

* + 1. Высветить всех заказчиков (таблица **Customers**), которые не имеют ни одного заказа (подзапрос по таблице Orders).

Использовать коррелированный SELECT и оператор EXISTS.