

# Задание к модулю SQL Basic.

## Общее

Для работы мы будем использовать базу Northwind, исходный код которой можно найти в материалах, в файле instnwnd.sql.

## Задание 1. Оработка SQL запросов на поиск и фильтрацию

Результаты данного задания можно оформить или отдельными sql-скриптами, или добавить их в SSDT проект, который создать из файла instnwnd.sql (только не забывайте, что такие скрипты не нужно включать в компиляцию проекта).

### Задание 1.1. Простая фильтрация данных.

1. Выбрать в таблице Orders заказы, которые были доставлены после 6 мая 1998 года (колонок ShippedDate) включительно и которые доставлены с ShipVia >= 2. Запрос должен возвращать только колонки OrderID, ShippedDate и ShipVia.
2. Написать запрос, который выводит только недоставленные заказы из таблицы Orders. В результатах запроса возвращать для колонки ShippedDate вместо значений NULL строку 'Not Shipped' (использовать системную функцию CASE). Запрос должен возвращать только колонки OrderID и ShippedDate.
3. Выбрать в таблице Orders заказы, которые были доставлены после 6 мая 1998 года (ShippedDate) не включая эту дату или которые еще не доставлены. В запросе должны возвращаться только колонки OrderID (переименовать в Order Number) и ShippedDate (переименовать в Shipped Date). В результатах запроса возвращать для колонки ShippedDate вместо значений NULL строку 'Not Shipped', для остальных значений возвращать дату в формате по умолчанию.

### Задание 1.2. 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. Возвращать колонки с именем пользователя и названием страны в результатах запроса. Упорядочить результаты запроса по имени заказчиков и по месту проживания.
2. Выбрать из таблицы Customers всех заказчиков, не проживающих в USA и Canada. Запрос сделать с помощью оператора IN. Возвращать колонки с именем пользователя и названием страны в результатах запроса. Упорядочить результаты запроса по имени заказчиков.
3. Выбрать из таблицы Customers все страны, в которых проживают заказчики. Страна должна быть упомянута только один раз и список отсортирован по убыванию. Не использовать предложение GROUP BY. Возвращать только одну колонку в результатах запроса.

### Задание 1.3. 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 возвращать только колонку OrderID.
2. 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 возвращать только колонки CustomerID и Country и отсортирован по Country.

3. Выбрать всех заказчиков из таблицы Customers, у которых название страны начинается на буквы из диапазона b и g, не используя оператор BETWEEN.

#### Задание 1.4. Использование оператора LIKE

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

## Задание 2. Обработка SQL запросов на объединение таблиц и агрегацию

#### Задание 2.1. Использование агрегатных функций (SUM, COUNT)

1. Найти общую сумму всех заказов из таблицы Order Details с учетом количества закупленных товаров и скидок по ним. Результатом запроса должна быть одна запись с одной колонкой с названием колонки 'Totals'.
2. По таблице Orders найти количество заказов, которые еще не были доставлены (т.е. в колонке ShippedDate нет значения даты доставки). Использовать при этом запросе только оператор COUNT. Не использовать предложения WHERE и GROUP.
3. По таблице Orders найти количество различных покупателей (CustomerID), сделавших заказы. Использовать функцию COUNT и не использовать предложения WHERE и GROUP.

#### Задание 2.2. Соединение таблиц, использование агрегатных функций и предложений GROUP BY и HAVING

1. По таблице Orders найти количество заказов с группировкой по годам. В результатах запроса надо возвращать две колонки с названиями Year и Total. Написать проверочный запрос, который вычисляет количество всех заказов.
2. По таблице Orders найти количество заказов, сделанных каждым продавцом. Заказ для указанного продавца – это любая запись в таблице Orders, где в колонке EmployeeID задано значение для данного продавца. В результатах запроса надо возвращать колонку с именем продавца (Должно высвечиваться имя полученное конкатенацией LastName & FirstName. Эта строка LastName & FirstName должна быть получена отдельным запросом в колонке основного запроса. Также основной запрос должен использовать группировку по EmployeeID.) с названием колонки 'Seller' и колонку с количеством заказов возвращать с названием 'Amount'. Результаты запроса должны быть упорядочены по убыванию количества заказов.
3. По таблице Orders найти количество заказов, сделанных каждым продавцом и для каждого покупателя. Необходимо определить это только для заказов, сделанных в 1998 году.
4. Найти покупателей и продавцов, которые живут в одном городе. Если в городе живут только один или несколько продавцов, или только один или несколько покупателей, то информация о таких покупателя и продавцах не должна попадать в результирующий набор. Не использовать конструкцию JOIN.
5. Найти всех покупателей, которые живут в одном городе.
6. По таблице Employees найти для каждого продавца его руководителя.

#### Задание 2.3. Использование JOIN

1. Определить продавцов, которые обслуживают регион 'Western' (таблица Region).

2. Выдать 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.

#### Задание 2.4. Использование подзапросов

1. Выдать всех поставщиков (колонок 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.
2. Выдать всех продавцов, которые имеют более 150 заказов. Использовать вложенный SELECT.
3. Выдать всех заказчиков (таблица Customers), которые не имеют ни одного заказа (подзапрос по таблице Orders). Использовать оператор EXISTS.

### Задание 3. Развертывание базы и обновлений

Ваша задача – подготовить к выпуску несколько версий проекта Northwind Extended, а точнее – базу данных проекта. Исходный проект основан на оригинальной базе Northwind (исходный код которой можно найти в материалах, в файле instnwnd.sql).

В рамках Northwind Extended предполагается выпустить следующие 3 версии:

- **Версия 1.0.** Основана на оригинальной базе Northwind
- **Версия 1.1.** Добавляет таблицу данных кредитных карт сотрудников: номер карты, дата истечения, имя card holder, ссылку на сотрудника, ...
- **Версия 1.3.** Добавляет следующие минорные изменения относительно 1.1:
  - Переименование Region в Regions
  - Добавление в таблицу клиентов даты основания

Вся работа выполняется на SQL Server любой доступной редакции.

#### Задание 3.1. Использование Alter Scripts

Создайте 2 sql-скрипта, которые выполняют обновление базы по версиям:

- 1.0 -> 1.1
- 1.1 -> 1.3

При выполнении задания добиться того, чтобы скрипты можно было накатывать многократно (например, для случая ошибочного повторного обновления) без ошибок.

#### Задание 3.2. Использование SSDT

Импортируйте исходную базу Northwind (из базы или скрипта instnwnd.sql) в проект базы данных для Visual Studio. **Импортируйте только метаданные, данные переносить не надо!**

Представьте 3 версии проекта на базе SSDT с изменениями, как описано выше.

Убедитесь, что при использовании SSDT вы можете обновить версии как последовательно, так и с пропуском версии. А также, что работает многократное обновление.

#### Задание 3.3. Вставка данных при деплое

Дополните SSDT с проектом с базой Northwind версии 1.3 post deploy скриптом вставки данных для таблиц

- Categories
- Suppliers
- Products

Можно взять данные из исходного скрипта `instnwnd.sql` (можно брать не все данные, а первые 5-10 записей)

При этом ваш проект должен позволять накатывать данные несколько раз (при условии, что в этих и других таблицах не появилось новых данных).