

# Отчёт к модулю SQL Basic

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

### Задание 1.1. Простая фильтрация данных

```
--Выбрать в таблице Orders заказы, которые были доставлены после 6 мая 1998 года (колодка ShippedDate)
--включительно и которые доставлены с ShipVia >= 2. Запрос должен возвращать только колонки OrderID,
--ShippedDate и ShipVia.

DECLARE
    @date DATETIME := Convert(DATETIME, '1998-05-06'),
    @shipVia INT := 2;

SELECT
    OrderT.[OrderID] AS 'OrderID',
    OrderT.[ShippedDate] AS 'ShippedDate',
    OrderT.[ShipVia] AS 'ShipVia'
FROM [dbo].[Orders] OrderT
WHERE OrderT.[ShippedDate] >= @date AND OrderT.[ShipVia] >= @shipVia;
```

	OrderID	ShippedDate	ShipVia
1	11063	1998-05-06 00:00:00.000	2
2	11067	1998-05-06 00:00:00.000	2
3	11069	1998-05-06 00:00:00.000	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.

SELECT
    OrdersT.[OrderID] AS 'OrderID',
    CASE
        WHEN OrdersT.[ShippedDate] IS NULL
        THEN 'Not shipped' END
    AS 'ShippedDate'
FROM [dbo].[Orders] OrdersT
WHERE OrdersT.[ShippedDate] IS NULL;
```

	OrderID	ShippedDate
1	11008	Not shipped
2	11019	Not shipped
3	11039	Not shipped
4	11040	Not shipped
5	11045	Not shipped
6	11051	Not shipped
7	11054	Not shipped
8	11058	Not shipped
9	11059	Not shipped
10	11061	Not shipped

	OrderID	ShippedDate
11	11062	Not shipped
12	11065	Not shipped
13	11068	Not shipped
14	11070	Not shipped
15	11071	Not shipped
16	11072	Not shipped
17	11073	Not shipped
18	11074	Not shipped
19	11075	Not shipped
20	11076	Not shipped
21	11077	Not shipped

```

--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.

DECLARE
    @date DATETIME = CONVERT(DATETIME, '1998-05-06'),
    @DEFAULT_DATETIME_FORMAT INT = 0;

SELECT
    OrdersT.[OrderID] AS 'Order Number',
    CASE
        WHEN OrdersT.[ShippedDate] IS NULL
        THEN 'Not shipped'
        ELSE CONVERT(VARCHAR(30), OrdersT.[ShippedDate], @DEFAULT_DATETIME_FORMAT)
    END AS 'Shipped Date'
FROM [dbo].[Orders] OrdersT
WHERE OrdersT.[ShippedDate] > @date OR OrdersT.[ShippedDate] IS NULL;

```

	Order Number	Shipped Date
1	11008	Not shipped
2	11019	Not shipped
3	11039	Not shipped
4	11040	Not shipped
5	11045	Not shipped
6	11051	Not shipped
7	11054	Not shipped
8	11058	Not shipped
9	11059	Not shipped
10	11061	Not shipped
11	11062	Not shipped
12	11065	Not shipped
13	11068	Not shipped
14	11070	Not shipped
15	11071	Not shipped
16	11072	Not shipped
17	11073	Not shipped
18	11074	Not shipped
19	11075	Not shipped
20	11076	Not shipped
21	11077	Not shipped

## Задание 1.2. Использование операторов IN, DISTINCT, ORDER BY, NOT

```

--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.

SELECT
    CustomersT.[ContactName] AS 'Contact Name',
    CustomersT.[Country] AS 'Country'
FROM [dbo].[Customers] CustomersT
WHERE CustomersT.[Country] IN ('USA', 'Canada')
ORDER BY CustomersT.[ContactName], CustomersT.[Address];

```

	Contact Name	Country
1	Art Braunschweiger	USA
2	Elizabeth Lincoln	Canada
3	Fran Wilson	USA
4	Helvetius Nagy	USA
5	Howard Snyder	USA
6	Jaime Yorres	USA
7	Jean Fresniere	Canada
8	John Steel	USA
9	Jose Pavarotti	USA
10	Karl Jablonski	USA
11	Liu Wong	USA
12	Liz Nixon	USA
13	Paula Wilson	USA
14	Rene Phillips	USA
15	Yoshi Latimer	USA
16	Yoshi Tannamuri	Canada

```

--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.

SELECT
    ....CustomersT.[ContactName] ....AS 'Contact Name',
    ....CustomersT.[Country] ... AS 'Country'
FROM [dbo].[Customers] CustomersT
WHERE CustomersT.[Country] NOT IN ('USA', 'Canada')
ORDER BY CustomersT.[ContactName];

```

	Contact Name	Country
1	Alejandra Camino	Spain
2	Alexander Feuer	Germ...
3	Ana Trujillo	Mexico
4	Anabela Domin...	Brazil
5	Andre Fonseca	Brazil
6	Ann Devon	UK
7	Annette Roulet	France
8	Antonio Moreno	Mexico
9	Aria Cruz	Brazil
10	Bernardo Batista	Brazil
11	Carine Schmitt	France
12	Carlos Gonzalez	Vene...
13	Carlos Hernandez	Vene...
14	Catherine Dewey	Belgium
15	Christina Berglu...	Swed...
16	Daniel Tonini	France
17	Diego Roel	Spain
18	Dominique Perier	France
19	Eduardo Saave...	Spain
20	Elizabeth Brown	UK
21	Felipe Izquierdo	Vene...
22	Francisco Chang	Mexico
23	Frederique Cite...	France
24	Georg Pippis	Austria
25	Giovanni Rovelli	Italy

26	Guillermo Fema...	Mexico
27	Hanna Moos	Germ...
28	Hari Kumar	UK
29	Helen Bennett	UK
30	Henriette Pfalz...	Germ...
31	Horst Kloss	Germ...
32	Isabel de Castro	Portu...
33	Janete Limeira	Brazil
34	Janine Labruno	France
35	Jonas Bergulfen	Norway
36	Jose Pedro Fre...	Spain
37	Jytte Petersen	Denm...
38	Karin Josephs	Germ...
39	Laurence Lebi...	France
40	Lino Rodriguez	Portu...
41	Lucia Carvalho	Brazil
42	Manuel Pereira	Vene...
43	Maria Anders	Germ...
44	Maria Larsson	Swed...
45	Marie Bertrand	France
46	Mario Pontes	Brazil
47	Martin Sommer	Spain
48	Martine Rance	France
49	Mary Saveley	France
50	Matti Karttunen	Finland

51	Maurizio Moroni	Italy
52	Michael Holz	Switz...
53	Miguel Angel P...	Mexico
54	Palle Ibsen	Denm...
55	Paolo Accorti	Italy
56	Pascale Cartrain	Belgium
57	Patricia McKenna	Ireland
58	Patricio Simpson	Argen...
59	Paul Henriot	France
60	Paula Parente	Brazil
61	Pedro Afonso	Brazil
62	Peter Franken	Germ...
63	Philip Cramer	Germ...
64	Pirkko Koskitalo	Finland
65	Renate Messner	Germ...
66	Rita Muller	Germ...
67	Roland Mendel	Austria
68	Sergio Gutierrez	Argen...
69	Simon Crowther	UK
70	Sven Ottlieb	Germ...
71	Thomas Hardy	UK
72	Victoria Ashworth	UK
73	Yang Wang	Switz...
74	Yvonne Monca...	Argen...
75	Zbyszek Piestrz...	Poland

```
--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.

SELECT DISTINCT
    CustomersT.[Country] AS 'Country'
FROM [dbo].[Customers] CustomersT
ORDER BY CustomersT.[Country] DESC;
```

	Country		
1	Venezuela	11	Italy
2	USA	12	Ireland
3	UK	13	Germany
4	Switzerla...	14	France
5	Sweden	15	Finland
6	Spain	16	Denmark
7	Portugal	17	Canada
8	Poland	18	Brazil
9	Norway	19	Belgium
10	Mexico	20	Austria
		21	Argentina

### Задание 1.3. Использование оператора BETWEEN, DISTINCT

```
--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.

SELECT DISTINCT
    OrderDetailsT.[OrderId] AS 'OrderId'
FROM [dbo].[Order Details] OrderDetailsT
WHERE OrderDetailsT.[Quantity] BETWEEN 3 AND 10;
```

	OrderId		
1	10248	382	11064
2	10249	383	11065
3	10250	384	11066
4	10251	385	11067
5	10257	386	11068
6	10258	387	11071
7	10259	388	11072
8	10268	389	11073
9	10272	390	11075
10	10274	391	11076
		392	11077

```
--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.

SELECT
    CustomersT.[CustomerId] AS 'CustomerId',
    CustomersT.[Country] AS 'Country'
FROM [dbo].[Customers] CustomersT
WHERE SUBSTRING(CustomersT.[Country], 1, 1) BETWEEN 'b' AND 'g'
ORDER BY CustomersT.[Country];
```

	CustomerId	Country		CustomerId	Country		CustomerId	Country
1	MAISD	Belgium	11	WELLI	Brazil	31	KOENE	Germ...
2	SUPRD	Belgium	12	LAUGB	Canada	32	LEHMS	Germ...
3	RICAR	Brazil	13	BOTTM	Canada	33	ALFKI	Germ...
4	TRADH	Brazil	14	MEREP	Canada	34	BLAUS	Germ...
5	QUEDE	Brazil	15	VAFFE	Denm...	35	DRACD	Germ...
6	QUEEN	Brazil	16	SIMOB	Denm...	36	QUICK	Germ...
7	COMMI	Brazil	17	WILMK	Finland	37	MORGK	Germ...
8	FAMIA	Brazil	18	WARTH	Finland	38	OTTIK	Germ...
9	GOURL	Brazil	19	SPECD	France	39	WANDK	Germ...
10	HANAR	Brazil	20	VICTE	France	40	TOMSP	Germ...

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

SELECT
    ....CustomersT.[CustomerId] AS 'CustomerId',
    ....CustomersT.[Country] AS 'Country'
FROM [dbo].[Customers] CustomersT WHERE SUBSTRING(CustomersT.[Country], 1, 1) IN ('b', 'c', 'd', 'e', 'f', 'g')
ORDER BY CustomersT.[Country];

SELECT
    ....CustomersT.[CustomerId] AS 'CustomerId',
    ....CustomersT.[Country] AS 'Country'
FROM [dbo].[Customers] CustomersT WHERE SUBSTRING(Country, 1, 1) >= 'b' AND SUBSTRING(Country, 1, 1) <= 'g'
ORDER BY CustomersT.[Country];
```

	CustomerId	Country
1	MAISD	Belgium
2	SUPRD	Belgium
3	RICAR	Brazil
4	TRADH	Brazil
5	QUEDE	Brazil
6	QUEEN	Brazil
7	COMMI	Brazil
8	FAMIA	Brazil

	CustomerId	Country
11	WELLI	Brazil
12	LAUGB	Canada

#### Задание 1.4. Использование оператора LIKE

```
--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.

SELECT ProductsT.[ProductName] AS 'Product name'
FROM [dbo].[Products] ProductsT
WHERE ProductsT.[ProductName] LIKE '%cho_olate%';
```

	Product name
1	Teatime Chocolate Biscuits

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

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

```
--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'.

SELECT SUM(OrderDetails.[Quantity] * OrderDetails.[UnitPrice] * (1 - OrderDetails.[Discount])) AS 'Totals'
FROM [dbo].[Order Details] OrderDetails;
```

	Totals
1	1265793,03961849

```
--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.

SELECT COUNT(*) -- COUNT(OrdersT.[ShippedDate]) AS 'Not delivered count'
FROM [dbo].[Orders] OrdersT;
```

	Not delivered count
1	21

```
--По таблице Orders найти количество различных покупателей (CustomerId), сделавших заказы.
--Использовать функцию COUNT и не использовать предложения WHERE и GROUP.

SELECT COUNT(DISTINCT OrdersT.[CustomerId]) AS 'Distinct customers count'
FROM [dbo].[Orders] OrdersT;
```

	Distinct customers count
1	89

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

```
--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.

SELECT
    YEAR(OrdersT.[OrderDate]) AS 'Order Year',
    COUNT(OrdersT.[OrderId]) AS 'Orders count'
FROM [dbo].[Orders] OrdersT
GROUP BY YEAR(OrdersT.[OrderDate]);
```

	Order Year	Orders count
1	1998	270
2	1996	152
3	1997	408

```

--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.

SELECT
  (SELECT CONCAT(EmployeesT.[LastName], ' ', EmployeesT.[FirstName])
   FROM [dbo].[Employees] EmployeesT
   WHERE EmployeesT.[EmployeeID] = OrdersT.[EmployeeID])
  AS 'Seller',
  COUNT(OrdersT.[OrderId]) AS 'Amount'
FROM [dbo].[Orders] OrdersT
GROUP BY OrdersT.[EmployeeID]
ORDER BY 'Amount' DESC;

```

	Seller	Amount
1	Peacock Margaret	156
2	Leverling Janet	127
3	Davolio Nancy	123
4	Callahan Laura	104
5	Fuller Andrew	96
6	King Robert	72
7	Suyama Michael	67
8	Dodsworth Anne	43
9	Buchanan Steven	42

```

--По таблице Orders найти количество заказов, сделанных каждым продавцом и для каждого покупателя.
--Необходимо определить это только для заказов, сделанных в 1998 году.

DECLARE
  @year INT = 1998;

SELECT
  OrdersT.[EmployeeID] AS 'EmployeeId',
  OrdersT.[CustomerId] AS 'CustomerId',
  COUNT(OrdersT.[OrderId]) AS 'Amount'
FROM [dbo].[Orders] OrdersT
WHERE YEAR(OrdersT.[OrderDate]) = @year
GROUP BY OrdersT.[EmployeeID], OrdersT.[CustomerId];

```

	EmployeeId	CustomerId	Amount
1	1	ALFKI	2
2	3	ALFKI	1
3	4	ANATR	1
4	3	ANTON	1
5	4	AROUT	2
6	9	AROUT	2
7	3	BERGS	1
8	4	BERGS	1
9	5	BERGS	1
10	8	BERGS	1

...

208	3	WHITC	1
209	4	WHITC	1
210	7	WHITC	1
211	1	WILMK	1
212	2	WILMK	1
213	3	WILMK	1
214	4	WILMK	1
215	4	WOLZA	2
216	5	WOLZA	1
217	8	WOLZA	1



```
--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 покупателях и
--продавцах не должна попадать в результирующий набор.
--Не использовать конструкцию JOIN
```

```
SELECT
  CustomersT.[CustomerId] AS 'CustomerId',
  EmployeesT.[EmployeeID] AS 'EmployeeId',
  CustomersT.[City] AS 'City'
FROM [dbo].[Customers] CustomersT
CROSS APPLY (SELECT EmployeesT.[EmployeeId]
  FROM [dbo].[Employees] EmployeesT
  WHERE EmployeesT.[City] = CustomersT.[City]) EmployeesT;
```

	CustomerId	EmployeeId	City
1	WHITC	1	Seattle
2	TRAIH	3	Kirkland
3	AROUT	5	London
4	BSBEV	5	London
5	CONSH	5	London
6	EASTC	5	London
7	NORTS	5	London
8	SEVES	5	London
9	AROUT	6	London
10	BSBEV	6	London

18	EASTC	7	London
19	NORTS	7	London
20	SEVES	7	London
21	WHITC	8	Seattle
22	AROUT	9	London
23	BSBEV	9	London
24	CONSH	9	London
25	EASTC	9	London
26	NORTS	9	London
27	SEVES	9	London

```
--Найти всех покупателей, которые живут в одном городе
```

```
SELECT
  CustomersL.[CustomerID] AS 'Customer id',
  CustomersR.[CustomerID] AS 'Neighbor id',
  CustomersL.[City] AS 'City'
FROM [dbo].[Customers] CustomersL
  LEFT JOIN [dbo].[Customers] CustomersR
  ON CustomersL.[CustomerID] <> CustomersR.[CustomerID] AND CustomersL.[City] = CustomersR.[City]
ORDER BY CustomersL.[CustomerID];
```

	Customer id	Neighbor id	City
1	ALFKI	NULL	Berlin
2	ANATR	ANTON	Mexico D.F.
3	ANATR	CENTC	Mexico D.F.
4	ANATR	PERIC	Mexico D.F.
5	ANATR	TORTU	Mexico D.F.
6	ANTON	ANATR	Mexico D.F.
7	ANTON	CENTC	Mexico D.F.
8	ANTON	PERIC	Mexico D.F.
9	ANTON	TORTU	Mexico D.F.
10	AROUT	BSBEV	London

138	TRAIH	NULL	Kirkland
139	VAFFE	NULL	Aarhus
140	VICTE	NULL	Lyon
141	VINET	NULL	Reims
142	WANDK	NULL	Stuttgart
143	WARTH	NULL	Oulu
144	WELLI	NULL	Resende
145	WHITC	NULL	Seattle
146	WILMK	NULL	Helsinki
147	WOLZA	NULL	Warszawa



--По таблице Employees найти для каждого продавца его руководителя.

```
SELECT
  → EmployeesT.[EmployeeID] → AS 'EmployeeID',
  → EmployeesT.[FirstName] → AS 'Seller name',
  → (SELECT ManagersT.[FirstName]
  → → FROM [dbo].[Employees] ManagersT
  → → WHERE ManagersT.[EmployeeID] = EmployeesT.[ReportsTo])
  → AS 'Manager'
FROM [dbo].[Employees] EmployeesT;
```

	EmployeeID	Seller name	Manager
1	1	Nancy	Andrew
2	2	Andrew	NULL
3	3	Janet	Andrew
4	4	Margaret	Andrew
5	5	Steven	Andrew
6	6	Michael	Steven
7	7	Robert	Steven
8	8	Laura	Andrew
9	9	Anne	Steven

### Задание 2.3. Использование JOIN

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

```
SELECT DISTINCT
  → EmployeesT.[EmployeeId] → AS 'EmployeeId',
  → EmployeesT.[FirstName] → AS 'First name'
FROM [dbo].[Employees] EmployeesT
  → → INNER JOIN [dbo].[EmployeeTerritories] EmployeeTerritoriesT
  → → → ON EmployeesT.[EmployeeID] = EmployeeTerritoriesT.[EmployeeID]
  → → INNER JOIN [dbo].[Territories] TerritoriesT
  → → → ON EmployeeTerritoriesT.[TerritoryID] = TerritoriesT.[TerritoryID]
  → → INNER JOIN [dbo].[Region] RegionT
  → → → ON RegionT.[RegionID] = TerritoriesT.[RegionID]
WHERE RegionT.[RegionDescription] = 'Western';
```

	EmployeeId	First name
1	6	Michael
2	7	Robert

```
--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.

SELECT CustomersT.[ContactName] AS 'ContactName',
       COUNT(OrdersT.[OrderId]) AS 'OrdersCount'
FROM [dbo].[Customers] CustomersT
     LEFT JOIN [dbo].[Orders] OrdersT
       ON CustomersT.[CustomerId] = OrdersT.[CustomerId]
GROUP BY CustomersT.[CustomerId], CustomersT.[ContactName]
ORDER BY 'OrdersCount';
```

	ContactName	OrdersCount
1	Diego Roel	0
2	Marie Bertrand	0
3	Francisco Chang	1
4	Manuel Pereira	2
5	John Steel	2
6	Yoshi Tannamuri	3
7	Simon Crowther	3
8	Carine Schmitt	3
9	Elizabeth Brown	3
10	Martin Sommer	3

82	Renate Messner	15
83	Laurence Lebihan	17
84	Christina Berglund	18
85	Carlos Hernandez	18
86	Paula Wilson	18
87	Patricia McKenna	19
88	Maria Larsson	19
89	Horst Kloss	28
90	Roland Mendel	30
91	Jose Pavarotti	31

#### Задание 2.4. Использование подзапросов

```
--Выдать 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

SELECT SuppliersT.[CompanyName] AS 'CompanyName'
FROM [dbo].[Suppliers] SuppliersT
WHERE SuppliersT.[SupplierID] IN (SELECT ProductsT.[SupplierID]
                                  FROM [dbo].[Products] ProductsT
                                  WHERE ProductsT.[UnitsInStock] = 0);
```

	CompanyName
1	New Orleans Cajun Delights
2	Pavlova, Ltd.
3	Plutzer Lebensmittelgroßmarkte AG
4	Fornaggi Fortini s.r.l.
5	G'day, Mate

```
--Выдать всех продавцов, которые имеют более 150 заказов. Использовать вложенный SELECT  
SELECT EmployeesT.[EmployeeID] AS 'EmployeeID'  
FROM [dbo].[Employees] EmployeesT  
WHERE (SELECT COUNT(OrdersT.[OrderID])  
→ FROM [dbo].[Orders] OrdersT  
→ WHERE OrdersT.[EmployeeID] = EmployeesT.[EmployeeID]) > 150;
```

	EmployeeID
1	4

```
--Выдать всех заказчиков (таблица Customers), которые не имеют ни одного заказа  
--(подзапрос по таблице Orders). Использовать оператор EXISTS  
SELECT CustomersT.[CustomerId]  
FROM [dbo].[Customers] CustomersT  
WHERE NOT EXISTS (SELECT OrdersT.[OrderID]  
→ FROM [dbo].[Orders] OrdersT  
→ WHERE OrdersT.[CustomerId] = CustomersT.[CustomerId]);
```

	CustomerId
1	FISSA
2	PARIS

## Задание 3. Развертывание базы и обновлений

### Задание 3.1. Использование Alter Scripts

```
--Версия 1.1
--Добавляет таблицу данных кредитных карт сотрудников:
--номер карты, дата истечения, имя card holder, ссылку на сотрудника,

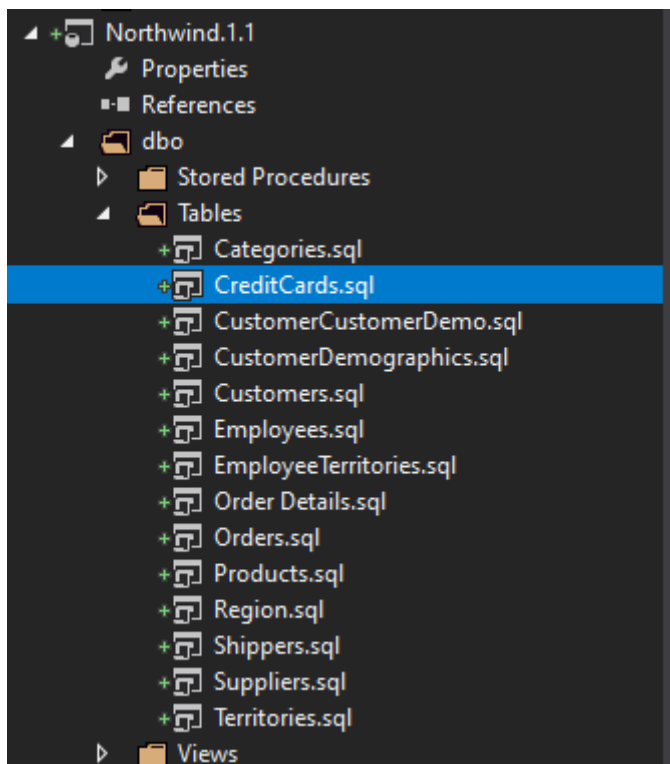
IF NOT EXISTS (SELECT * FROM SYS.TABLES SysTables WHERE SysTables.[Name] = 'CreditCards')
BEGIN
    CREATE TABLE [dbo].[CreditCards](
        [CreditCardId] INT NOT NULL PRIMARY KEY IDENTITY(1,1),
        [ExpirationDate] DATETIME DEFAULT(NULL),
        [CardHolderName] VARCHAR(200) NOT NULL,
        [EmployeeId] INT REFERENCES [dbo].[Employees]([EmployeeID]);
END
```

```
--Версия 1.3
--Добавляет следующие минорные изменения относительно 1.1:
--Переименование Region в Regions
--Добавление в таблицу клиентов даты основания

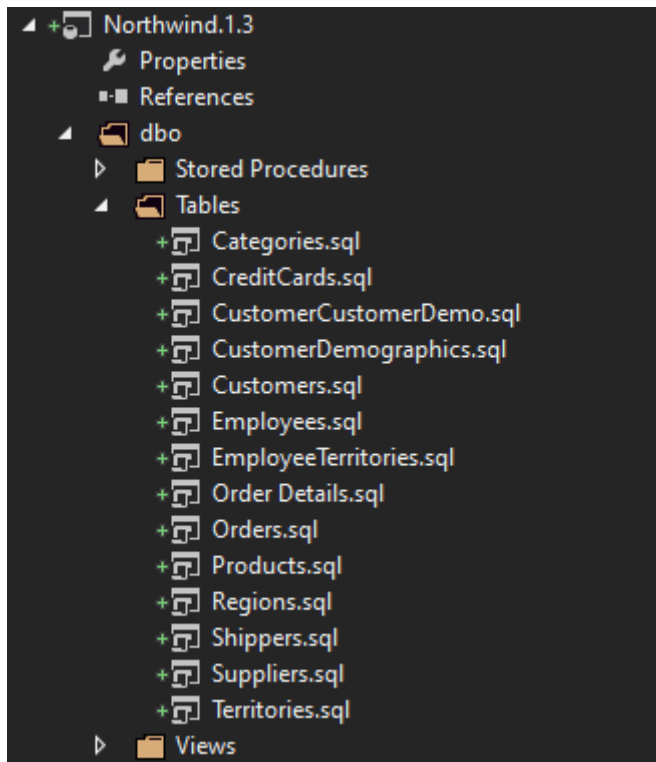
IF EXISTS (SELECT * FROM SYS.TABLES SystemTables WHERE SystemTables.[Name] = 'Region')
BEGIN
    EXEC SP_RENAME 'dbo.[Region]', 'Regions';
END

IF NOT EXISTS (SELECT * FROM SYS.COLUMNS SystemColumns
    WHERE SystemColumns.[OBJECT_ID] = OBJECT_ID(N'[dbo].[Customers]') AND Name = 'FoundationDate')
BEGIN
    ALTER TABLE [dbo].[Customers]
    ADD [FoundationDate] DATETIME
END
```

## Задание 3.2. Использование SSDT



Script File: CreditCards.sql				
	Name	Data Type	Allow Nulls	Default
	CreditCardId	int	<input type="checkbox"/>	
	ExpirationDate	datetime	<input checked="" type="checkbox"/>	(NULL)
	CardHolderName	varchar(200)	<input type="checkbox"/>	
	EmployeeId	int	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	



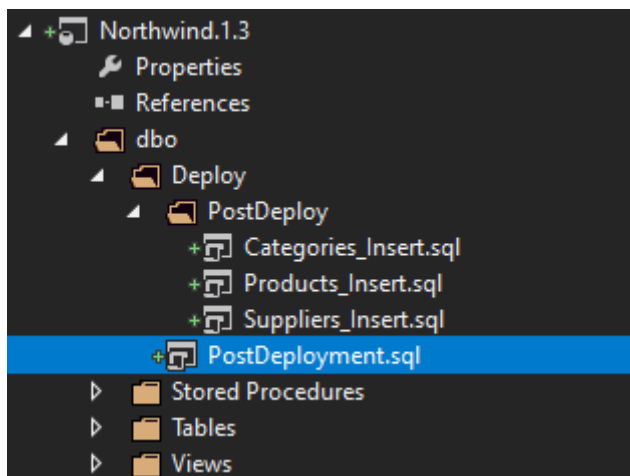
Script File: Regions.sql

	Name	Data Type	Allow Nulls	Default
PK	RegionID	int	<input type="checkbox"/>	
	RegionDescription	nchar(50)	<input type="checkbox"/>	
			<input type="checkbox"/>	

Script File: Customers.sql

	Name	Data Type	Allow Nulls	Default
PK	CustomerID	nchar(5)	<input type="checkbox"/>	
	CompanyName	nvarchar(40)	<input type="checkbox"/>	
	ContactName	nvarchar(30)	<input checked="" type="checkbox"/>	
	ContactTitle	nvarchar(30)	<input checked="" type="checkbox"/>	
	Address	nvarchar(60)	<input checked="" type="checkbox"/>	
	City	nvarchar(15)	<input checked="" type="checkbox"/>	
	Region	nvarchar(15)	<input checked="" type="checkbox"/>	
	PostalCode	nvarchar(10)	<input checked="" type="checkbox"/>	
	Country	nvarchar(15)	<input checked="" type="checkbox"/>	
	Phone	nvarchar(24)	<input checked="" type="checkbox"/>	
	Fax	nvarchar(24)	<input checked="" type="checkbox"/>	
	FoundationDate	datetime	<input checked="" type="checkbox"/>	
			<input type="checkbox"/>	

## Задание 3.3. Вставка данных при деплое



```
/*
Post-Deployment Script Template
-----
This file contains SQL statements that will be appended to the build script.
Use SQLCMD syntax to include a file in the post-deployment script.
Example: .....:r.\myfile.sql
Use SQLCMD syntax to reference a variable in the post-deployment script.
Example: .....:setvar TableName MyTable
.....SELECT * FROM [$(TableName)]
-----
*/

:r.\PostDeploy\Categories_Insert.sql
:r.\PostDeploy\Suppliers_Insert.sql
:r.\PostDeploy\Products_Insert.sql
```