**Практическая работа № 6. Реализация манипулирования данными и выборки данных из базы данных.**

**Программное обеспечение: Windows 8 и выше, Microsoft Office Word 2016 и выше, Microsoft SQL Server 2017 и выше, Microsoft SQL Server Management Studio 2017 и выше.**

**Цель работы:**

**На основании разработанной структуры базы данных с помощью языка DDL, произвести заполнение структуры базы данных и выборку данных из таблиц базы данных.**

**Время на выполнение практической работы:** 2 часа.

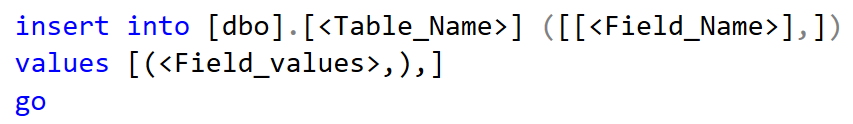
**Получение навыков:**

* **В работе группой команд DML;**
* **В отработке запросов на заполнение данными – Insert;**
* **В отработке запросов на изменение данных – Update;**
* **В отработке запросов на удаление данных – Delete;**
* **В отработке запросов на выборку данных из таблиц, с использованием объединения данных, фильтров, сортировки и объединения запросов – Select.**

**Теоретические сведения к выполнению практической работы:**

DML. Insert – Добавление данных.

Общая конструкция:



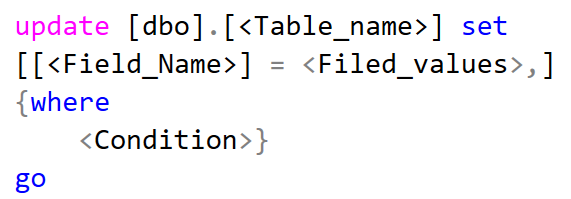
Правила создания:

1. В запросе указываются все поля, у которых есть свойство Not null;
2. Если поле автоинкриментно, его в запросе не указывают;
3. Количество присваиваемых значений совпадает с перечисленными полями, по количеству и порядку;
4. При добавлении данных внешнего ключа, необходимо обраться к значениям суррогатного ключа родительской таблицы;

В одном запросе может быть указано, несколько записей для добавления.

DML. Update – Изменение данных.

Общая конструкция:

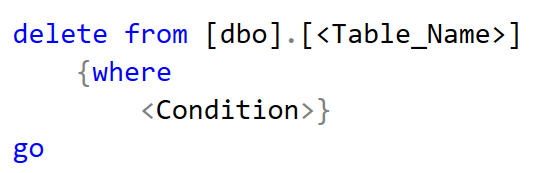


Правила создания:

1. Выборочное участие полей и значений в запросе;
2. Перед where запятая не ставится;
3. Запрос может быть записан без условия.

DML. Delete – Удаление данных.

Общая конструкция:



Правила создания:

1. Перед where запятая не ставится;
2. Запрос может быть записан без условия.

DML. Select – Выборка данных.

Ключевые области конструкций:

**Join construction** – область объединения данных из двух и более таблиц;

**Condition construction** – область описания условия к выборке данных;

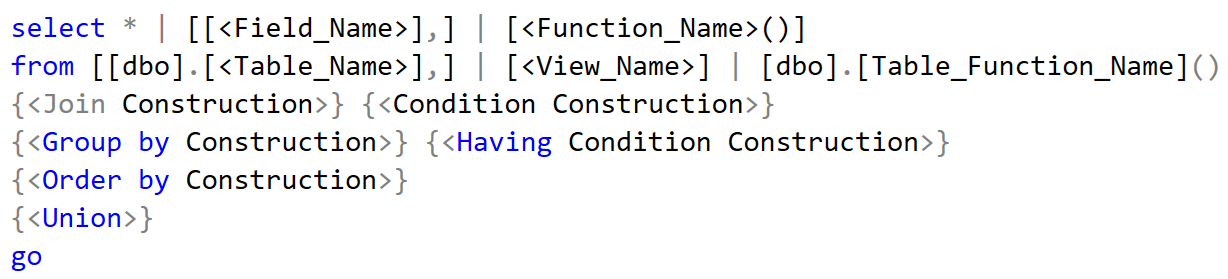
**Group by construction** – область группировки выполнения функций по количеству записей в столбце;

**Having condition construction** – область описания условия к результатам функций;

**Order by construction** – область описания порядка сортировки полей;

**Union** – область объединения двух и более запросов на выборку данных.

Общая конструкция:

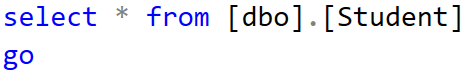


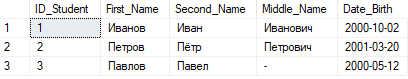
Правила создания:

1. \* - выборка всех полей из перечисленных таблиц;
2. Между Select и From должны обязательно быть перечислены поля.

Все поля из таблицы.

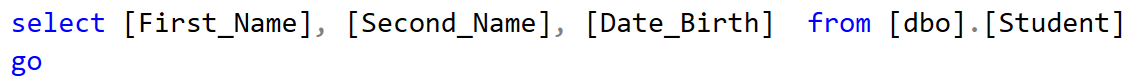
Пример:

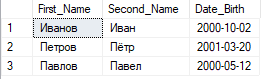




Перечисление полей.

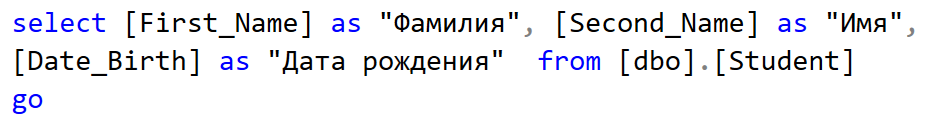
Пример:

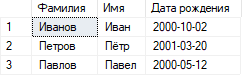




Псевдонимы полей – Alias.

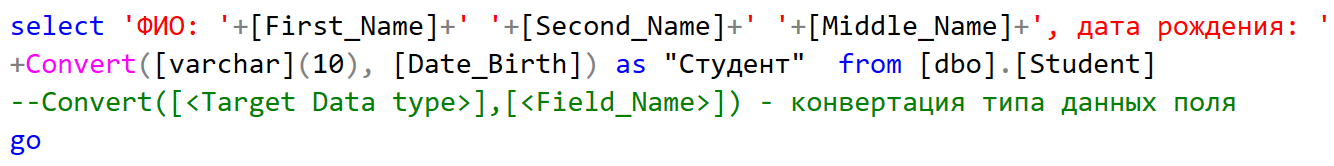
Пример:

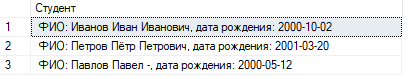




Конкатенация полей.

Пример:





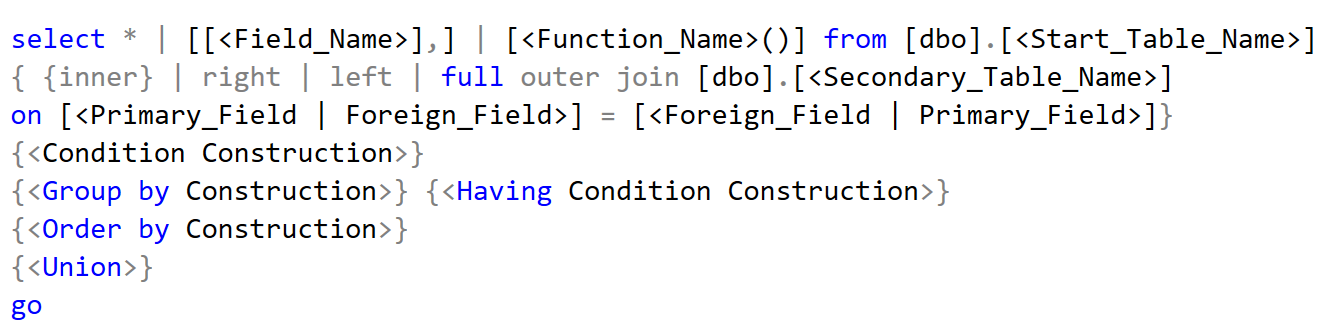
Общие правила, для простых запросов:

1. Все поля перечисляются через запятую;
2. Псевдонимы полей указываются через команду as в двойных кавычках, для каждого поля по отдельности;
3. Конкатенация полей указывается через +;
4. В конкатенируемых значениях, указываются шаблонные записи в одинарных кавычках.

DML. Select. Join – Объединение данных.

**Join** – позволяет объединять данные из двух и более таблиц, в одну результирующую таблицу, заменяя значения внешних ключей на, эквивалентные значения не ключевых атрибутов родительских таблиц.

Конструкция Join:



Правила написания join:

1. После from – указывается стартовая таблица с которой начинается объединение данных;
2. Конструкция join on – эквивалентна одной межтабличной связи;
3. Каждый из join on – не отделяется друг от друга запятыми;
4. Первичный (суррогатный) и внешний ключи в join - приравниваются друг к другу в любом порядке.

**{Inner} join** – представляет собой внутренне объединение данных из нескольких таблиц, при этом, будут выделены данные, которые учувствуют в межтабличной связи.

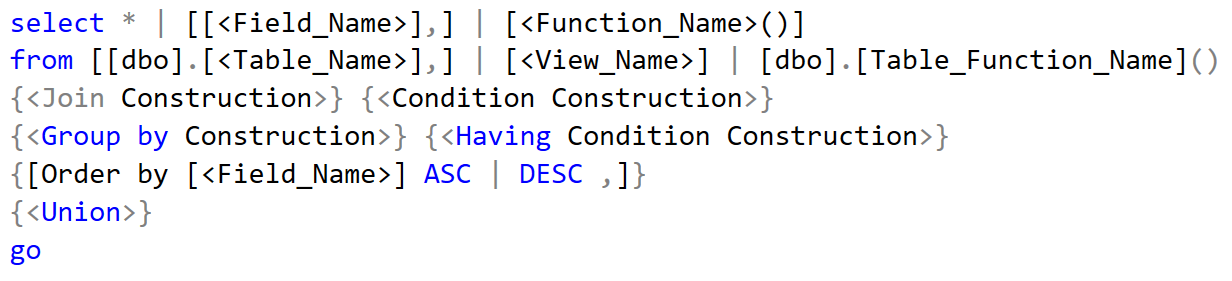
**Right join** – представляет собой объединение данных из нескольких таблиц, при этом, будут выделены данные, которые учувствуют в межтабличной связи и данные которые не задействованы из правой таблицы. Правая таблица указывается после join.

**Left join** – представляет собой объединение данных из нескольких таблиц, при этом, будут выделены данные, которые учувствуют в межтабличной связи и данные которые не задействованы из левой таблицы. Левая таблица указывается перед join.

**Full outer join** – представляет собой полное объединение данных из нескольких таблиц, при этом, будут выделены все данные, которые учувствуют и не учувствуют в межтабличных связях.

DML. Select – Сортировка данных.

Общая конструкция:



Правила создания:

1. Для прямой сортировки указывают ASC;
2. Для обратной сортировки указывают DESC;
3. В одном запросе могут быть отсортированы несколько полей в разных порядках;
4. Каждое поле отделяется друг от друга запятой.

DML. Select – Фильтрация данных.

Операторы сравнения:

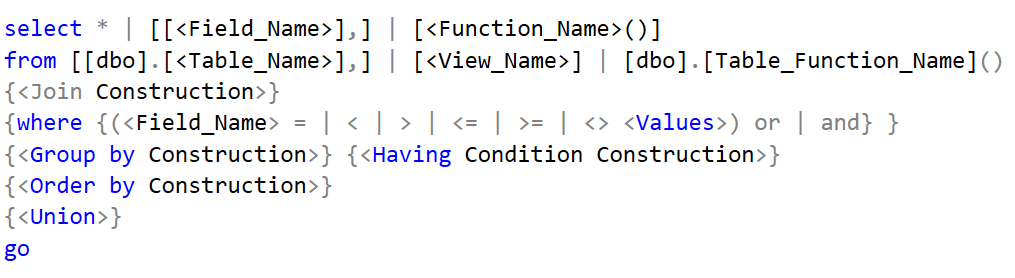
1. **=** - поле равно, указанному значению;
2. **<** - поле строго меньше указанного значения;
3. **>** - поле строго больше, указанного значения;
4. **<=** - поле меньше или равно, указанному значению;
5. **>=** - поле больше или равно, указанному значению;
6. **<>** - поле не равно, указанному значению.

Операторы объединения:

**1. Or** – логическое «Или», если хотя бы одно условие истинно, то все выражение истинное;

**2. And** – логическое «И», если все условия истинны, то все выражение истинное.

Общая конструкция:



Правила:

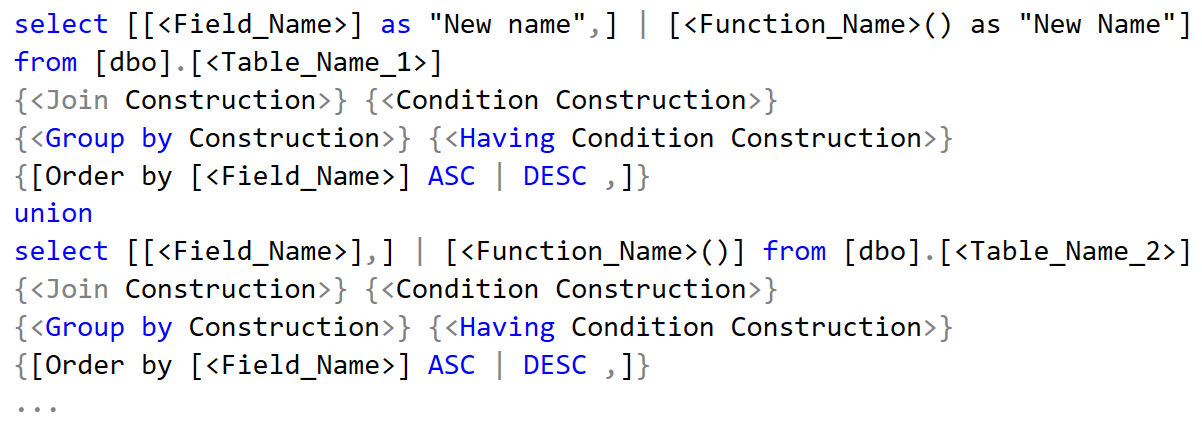
Каждое из частей условия, соединённых с помощью операторов сравнения, пишутся в круглых скобках.

DML. Select. Union – Объединение запросов.

**Union** – запрос позволяет объединять в одну результирующую таблицу от двух и более запросов на выборку данных. Основные правила:

1. Количество полей у запросов должно совпадать;
2. Типы полей должны совпадать;
3. Порядок полей должен совпадать;
4. Заголовки столбцов наследуются у первого запроса;
5. Использование специального символа \*, не допустимо.

Общая конструкция:



**Этапы выполнения:**

***Требования к заполнению данных:***

1. ***Каждая таблица должна иметь ровно столько записей, сколько было в 3НФ;***
2. ***При выборке данных из дочерних таблиц не должно быть значений первичных (суррогатных) и внешних ключей;***
3. ***Для фильтрации данных необходимо описать задачу (к 3-ём таблицам), на основании которой будет произведена выборка данных из базы данных***
4. ***Для 5-ти таблиц написать по 1 запросу: изменение и удаление данных.***
5. Общее представление манипулирования над данными в таблицах.

|  |  |
| --- | --- |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Type\_Product]  (  [ID\_Type\_Product] [int] not null identity(1,1),  [Name\_Type] [varchar] (50) not null  constraint [PK\_Type\_Product] primary key clustered  ([ID\_Type\_Product] ASC) on [PRIMARY],  constraint [UQ\_Name\_Type] unique ([Name\_Type])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Type\_Product] ([Name\_Type])  values ('Растворимый продукт'),  ('Овощь'),  ('Мясное изделье'),  ('Молочная продукция')  go |
| Запросы на выборку данных |
| Задача: Вывести название типа продукта, отсортировав название по возрастанию. |
| select [Name\_Type] as "Название типа продукта" from [dbo].[Type\_Product]  order by [Name\_Type] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Country]  (  [ID\_Country] [int] not null identity(1,1),  [Name\_Country] [varchar] (50) not null  constraint [PK\_Country] primary key clustered  ([ID\_Country] ASC) on [PRIMARY],  constraint [UQ\_Name\_Country] unique ([Name\_Country])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Country]  values ('Швейцария'),  ('Россия')  go |
| Запросы на выборку данных |
| Задача: Вывести название страны, отсортировав название по возрастанию. |
| select [Name\_Country] as "Название страны" from [dbo].[Country]  order by [Name\_Country] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Market\_Place]  (  [ID\_Market\_Place] [int] not null identity(1,1),  [Market\_Address] [varchar] (max) not null  constraint [PK\_Market\_Place] primary key clustered  ([ID\_Market\_Place] ASC) on [PRIMARY]  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Market\_Place] ([Market\_Address])  values ('ул. Чертановская, д.6, стр. 2'),  ('ул. Волжская, д.10, к.1')  go |
| Запросы на выборку данных |
| Задача: Вывести адреса магазинов, отсортировав адреса по возрастанию. |
| select [Market\_Address] as "Адрес магазнина" from [dbo].[Market\_Place]  order by [Market\_Address] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Type\_Supplier]  (  [ID\_Type\_Supplier] [int] not null identity(1,1),  [Name\_Supplier] [varchar] (30) not null  constraint [PK\_Type\_Supplier] primary key clustered  ([ID\_Type\_Supplier] ASC) on [PRIMARY],  constraint [UQ\_Name\_Supplier] unique ([Name\_Supplier])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Type\_Supplier] ([Name\_Supplier])  values ('ООО'),  ('НПАО'),  ('ПАО')  go |
| Запросы на выборку данных |
| Задача: Вывести типы фирм, отсортировав названия по убыванию. |
| select [Name\_Supplier] as "Тип фирмы" from [dbo].[Type\_Supplier]  order by [Name\_Supplier] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Payment\_Type]  (  [ID\_Payment\_Type] [int] not null identity(1,1),  [Name\_Payment\_Type] [varchar] (30) not null  constraint [PK\_Payment\_Type] primary key clustered  ([ID\_Payment\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Payment\_Type] unique ([Name\_Payment\_Type])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Payment\_Type] ([Name\_Payment\_Type])  values ('Наличный'),  ('Без наличный')  go |
| Запросы на выборку данных |
| Задача: Вывести типы оплаты, отсортировав названия по возрастанию. |
| select [Name\_Payment\_Type] as "Тип оплаты" from [dbo].[Payment\_Type]  order by [Name\_Payment\_Type] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Post]  (  [ID\_Post] [int] not null identity(1,1),  [Name\_Post] varchar (50) not null  constraint [PK\_Post] primary key clustered  ([ID\_Post] ASC) on [PRIMARY],  constraint [UQ\_Name\_Post] unique ([Name\_Post])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Post] ([Name\_Post])  values ('Продавец-Кассир'),  ('Старший продвец')  go |
| Запросы на выборку данных |
| Задача: Вывести названия должностей, отсортировав названия по возрастанию. |
| select [Name\_Post] as "Название должности" from [dbo].[Post]  order by [Name\_Post] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Employee]  (  [ID\_Employee] [int] not null identity(1,1),  [First\_Name\_Emp] [varchar] (30) not null,  [Second\_Name\_Emp] [varchar] (30) not null,  [Middle\_Name\_Emp] [varchar] (30) null default ('-'),  [Key\_Employee] [varchar] (10) not null,  [Login\_Employee] [varchar] (32) not null,  [Password\_Employee] [varchar] (32) not null  constraint [PK\_Employee] primary key clustered  ([ID\_Employee] ASC) on [PRIMARY],  constraint [UQ\_Login\_Employee] unique ([Login\_Employee]),  constraint [CH\_Key\_Employee] check (len([Key\_Employee]) = 10),  constraint [CH\_Key\_Employee\_Letter\_Upper] check ([Key\_Employee] like ('%[A-Z]%')),  constraint [CH\_Key\_Employee\_Letter\_Lower] check ([Key\_Employee] like ('%[a-z]%')),  constraint [CH\_Key\_Employee\_Symbols] check ([Key\_Employee] like ('%[!@#$%^&\*()]%')),  constraint [CH\_Login\_Employee] check (len([Login\_Employee])>=8),  constraint [CH\_Password\_Employee\_Upper] check ([Password\_Employee] like ('%[A-Z]%')),  constraint [CH\_Password\_Employee\_Letter\_Lower] check ([Password\_Employee] like ('%[a-z]%')),  constraint [CH\_Password\_Employee\_Symbols] check ([Password\_Employee] like ('%[!@#$%^&\*()]%'))  )  go | alter table [dbo].[Employee] drop constraint [CH\_Key\_Employee]  go  alter table [dbo].[Employee] drop constraint [CH\_Key\_Employee\_Letter\_Upper]  go  alter table [dbo].[Employee] drop constraint [CH\_Key\_Employee\_Letter\_Lower]  go  alter table [dbo].[Employee] drop constraint [CH\_Key\_Employee\_Symbols]  go  alter table [dbo].[Employee] alter column [Key\_Employee] [varchar] (10) null  go  alter table [dbo].[Employee] add default('-') for [Key\_Employee]  go |
| Запросы на добавление данных |
| insert into [dbo].[Employee] ([First\_Name\_Emp], [Second\_Name\_Emp], [Middle\_Name\_Emp], [Login\_Employee], [Password\_Employee])  values ('Павлов','Павел','Павлович','PavlovPP','Pa$$w0rd')  go  insert into [dbo].[Employee] ([First\_Name\_Emp], [Second\_Name\_Emp], [Middle\_Name\_Emp], [Key\_Employee],[Login\_Employee], [Password\_Employee])  values ('Дмитриев','Дмитрий','Дмитриевич','Rg63Frr!h','DmitriyDD','Pa$$w0rd')  go  insert into [dbo].[Employee] ([First\_Name\_Emp], [Second\_Name\_Emp], [Middle\_Name\_Emp], [Login\_Employee], [Password\_Employee])  values ('Олегов','Олег','Олегович','OlegovOO','Pa$$w0rd')  go |
| Запросы на выборку данных |
| Задача: Вывести ФИО сотрудников, ключи отмены, данные логина и пароля, отсортировав фамилии по возрастанию. |
| select [First\_Name\_Emp]+' '+[Second\_Name\_Emp]+' '+[Middle\_Name\_Emp] as "ФИО Сотрудника", [Key\_Employee] as "Ключ отмены", [Login\_Employee] as "Логин сотрудника",  [Password\_Employee] as "Пароль сотрудника" from [dbo].[Employee]  order by [First\_Name\_Emp] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Cash\_Machine]  (  [ID\_Cash\_Machine] [int] not null identity(1,1),  [Cash\_Machine\_Number] [varchar] (11) not null  constraint [PK\_Cash\_Machine] primary key clustered  ([ID\_Cash\_Machine] ASC) on [PRIMARY],  constraint [UQ\_Cash\_Machine\_Number] unique ([Cash\_Machine\_Number]),  constraint [CH\_Cash\_Machine\_Number] check ([Cash\_Machine\_Number] like('КА-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'))  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Cash\_Machine] ([Cash\_Machine\_Number])  values ('КА-00000001'),  ('КА-00000002'),  ('КА-00000003')  go |
| Запросы на выборку данных |
| Задача: Вывести номера кассовых аппаратов, отсортировав номера по возрастанию. |
| select [Cash\_Machine\_Number] as "Номер кассового аппарата" from [dbo].[Cash\_Machine]  order by [Cash\_Machine\_Number] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Buyer]  (  [ID\_Buyer] [int] not null identity(1,1),  [First\_Name\_Buyer] [varchar] (30) not null,  [Second\_Name\_Buyer] [varchar] (30) not null,  [Middle\_Name\_Buyer] [varchar] (30) null default ('-'),  [Login\_Buyer] [varchar] (32) not null,  [Password\_Buyer] [varchar] (32) not null  constraint [PK\_Buyer] primary key clustered  ([ID\_Buyer] ASC) on [PRIMARY],  constraint [UQ\_Login\_Buyer] unique ([Login\_Buyer]),  constraint [CH\_Login\_Buyer] check (len([Login\_Buyer])>=8),  constraint [CH\_Password\_Buyer\_Upper] check ([Password\_Buyer] like ('%[A-Z]%')),  constraint [CH\_Password\_Buyer\_Letter\_Lower] check ([Password\_Buyer] like ('%[a-z]%')),  constraint [CH\_Password\_Buyer\_Symbols] check ([Password\_Buyer] like ('%[!@#$%^&\*()]%'))  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Buyer] ([First\_Name\_Buyer], [Second\_Name\_Buyer], [Middle\_Name\_Buyer], [Login\_Buyer], [Password\_Buyer])  values ('Иванов','Иван','Иванович','IvanovII','Pa$$w0rd'),  ('Петров','Пётр','Петрович','PetrovPP','Pa$$w0rd'),  ('Андреев','Андрей','Андреевич','AndreevAA','Pa$$w0rd')  go |
| Запросы на выборку данных |
| Задача: Вывести ФИО покупателей, логины и пароли, отсортировав фамилии по возрастанию. |
| select [First\_Name\_Buyer]+' '+[Second\_Name\_Buyer]+' '+[Middle\_Name\_Buyer] as "ФИО Клиента", [Login\_Buyer] as "Логин покупателя", [Password\_Buyer] as "Пароль покупателя" from [dbo].[Buyer]  order by [First\_Name\_Buyer] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Product\_List]  (  [ID\_Product\_List] [int] not null identity(1,1),  [Product\_List\_Number] [varchar] (13) not null,  [Date\_Time\_Create\_PL] [datetime] null default(getdate())  constraint [PK\_Product\_List] primary key clustered  ([ID\_Product\_List] ASC) on [PRIMARY],  constraint [CH\_Product\_List\_Number] check ([Product\_List\_Number] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9]')  )  go | alter table [dbo].[Product\_List] add constraint [UQ\_Product\_List\_Number] unique ([Product\_List\_Number])  go |
| Запросы на добавление данных |
| insert into [dbo].[Product\_List] ([Product\_List\_Number])  values ('0000000001-21'),  ('0000000002-21'),  ('0000000003-21')  go |
| Запросы на выборку данных |
| Задача: Вывести номера товарных смет, даты и время создания, отсортировать дату и время по убыванию. |
| select [Product\_List\_Number] as "Номер товарной сметы", [Date\_Time\_Create\_PL] as "Дата и время создания" from [dbo].[Product\_List]  order by [Date\_Time\_Create\_PL] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Cancellation]  (  [ID\_Cancellation] [int] not null identity(1,1),  [Cancellation\_Number] [varchar] (16) not null,  [Date\_Create\_Cncll] [date] null default (getdate())  constraint [PK\_Cancellation] primary key clustered  ([ID\_Cancellation] ASC) on [PRIMARY],  constraint [CH\_Cancellation\_Number] check ([Cancellation\_Number] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9]-ОП'))  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Cancellation] ([Cancellation\_Number])  values ('0000000001-21-ОП'),  ('0000000004-21-ОП'),  ('0000000002-21-ОП'),  ('0000000003-21-ОП')  go |
| Запросы на выборку данных |
| Задача: Вывести номера отмены приёма и дату формирования, отсортировав дату формирования по убыванию. |
| select [Cancellation\_Number] as "Номер отмены приёма", [Date\_Create\_Cncll] as "Дата формирования" from [dbo].[Cancellation]  order by [Date\_Create\_Cncll] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Manufacturer]  (  [ID\_Manufacturer] [int] not null identity(1,1),  [Name\_Manufacturer] [varchar] (50) not null,  [Country\_ID] [int] not null  constraint [PK\_Manufacturer] primary key clustered  ([ID\_Manufacturer] ASC) on [PRIMARY],  constraint [UQ\_Name\_Manufacturer] unique ([Name\_Manufacturer]),  constraint [FK\_Country\_Manufacturer] foreign key ([Country\_ID])  references [dbo].[Country] ([ID\_Country])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Manufacturer] ([Name\_Manufacturer], [Country\_ID])  values ('Nestle',1),  ('1-ый овощь на р-не',2),  ('Мясницкий ряд',2),  ('Вимм-Биль-Дан',2)  go |
| Запросы на выборку данных |
| Задача: Вывести название производителя, объединив с название страны, отсортировав название страны по возрастанию, а название производителя по убыванию. |
| select [Name\_Manufacturer] as "Название производителя", [Name\_Country] as "Страна поизводителя" from [dbo].[Manufacturer]  inner join [dbo].[Country] on [Country\_ID] = [ID\_Country]  order by [Name\_Country] ASC, [Name\_Manufacturer] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Warehouse]  (  [ID\_Warehouse] [int] not null identity(1,1),  [Warehouse\_Adress] [varchar] (max) not null,  [Market\_Place\_ID] [int] not null  constraint [PK\_Warehouse] primary key clustered  ([ID\_Warehouse] ASC) on [PRIMARY],  constraint [FK\_Market\_Place\_Warehouse] foreign key ([Market\_Place\_ID])  references [dbo].[Market\_Place] ([ID\_Market\_Place])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Warehouse] ([Warehouse\_Adress], [Market\_Place\_ID])  values ('ул. Чертановская, д.6, стр. 3', 1),  ('ул. Чертановская, д.6, стр. 1', 1),  ('ул. Волжская, д.10, к.2', 2)  go |
| Запросы на выборку данных |
| Задача: Вывести адрес склада, объединив с адресом магазина, отсортировав адрес магазина и склада по возрастанию. |
| select [Warehouse\_Adress] as "Адрес склада", [Market\_Address] as "Адрес маказина" from [dbo].[Warehouse]  inner join [dbo].[Market\_Place] on [ID\_Market\_Place] = [Market\_Place\_ID]  order by [Warehouse\_Adress], [Market\_Address] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Suppler]  (  [ID\_Suppler] [int] not null identity(1,1),  [Type\_Suppler\_ID] [int] not null,  [Full\_Name\_Suppler] [varchar] (max) not null,  [Short\_Name\_Suppler] [varchar] (50) not null,  [Legal\_Address\_Suppler] [varchar] (max) not null,  [Physical\_Address\_Suppler] [varchar] (max) not null,  [Suppler\_E\_Mail] [varchar] (max) not null,  [Suppler\_Phone\_Number] [varchar] (17) not null,  [BIC] [varchar] (8) not null,  [ITN] [varchar] (12) not null,  [ARCoEaO] [varchar] (10) not null  constraint [PK\_Suppler] primary key clustered  ([ID\_Suppler] ASC) on [PRIMARY],  constraint [UQ\_Suppler\_Phone\_Number] unique ([Suppler\_Phone\_Number]),  constraint [UQ\_ITN] unique ([ITN]),  constraint [UQ\_ARCoEaO] unique ([ARCoEaO]),  constraint [CH\_Suppler\_E\_Mail] check ([Suppler\_E\_Mail] like ('%@%.%')),  constraint [CH\_Suppler\_Phone\_Number] check ([Suppler\_Phone\_Number] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),  constraint [CH\_BIC] check ([BIC] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')),  constraint [CH\_ITN] check ([ITN] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')),  constraint [CH\_ARCoEaO] check ([ARCoEaO] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')),  constraint [FK\_Type\_Suppler\_Suppler] foreign key ([Type\_Suppler\_ID])  references [dbo].[Type\_Supplier] ([ID\_Type\_Supplier])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Suppler] ([Type\_Suppler\_ID], [Full\_Name\_Suppler], [Short\_Name\_Suppler],  [Legal\_Address\_Suppler], [Physical\_Address\_Suppler], [Suppler\_E\_Mail], [Suppler\_Phone\_Number],  [BIC], [ITN], [ARCoEaO])  values (1, '"Первый поставщик"', '"П Пост"', 'г. Москва, 155367, ул. Волоколамская, д.6, стр. 5',  'г. Москва, 155367, ул. Волоколамская, д.6, стр. 5', '1postavschik@gmail.com', '+7(495)422-63-64',  '35674316', '396652867052', '4824096316'),  (2, '"Куда скажете туда и поедем"', '"КСТиП"', 'г. Москва, 156733, ул. Бибирево, д.17, к.6,',  'г. Москва, 173274, ул. Алтуфьево, д.47, стр.17', 'kuda\_tuda@gmail.com', '+7(499)526-88-18',  '29665123', '583349957123', '4995123385'),  (3, '"Быстро и надёжно"', '"БиН"', 'г. Санкт-Петербург, 4226432, ул. Новаторов, д.64, к.4',  'г. Москва, 198887, ул. Киевская, д.10, стр. 10', 'bin@gmail.com', '+7(999)542-77-87',  '83128904', '679333078641', '6722450097')  go |
| Запросы на выборку данных |
| Задача: Вывести тип фирмы, все реквизиты фирмы в одно поле, объединив данные с типом фирмы, отсортировав название типа фирмы по убыванию, а полное название фирмы по возрастанию. |
| select 'Тип фирмы: '+[Name\_Supplier]+', полное название: '+[Full\_Name\_Suppler]+', сокращённое название: '+[Short\_Name\_Suppler] as "Данные о постащике",  'Юридический адрес: '+[Legal\_Address\_Suppler]+', фактическтй адрес: '+[Physical\_Address\_Suppler]+'. Адрес электронной почты: '+[Suppler\_E\_Mail]+', контактный телефон: '+[Suppler\_Phone\_Number] as "Контактная информация",  'БИК: '+[BIC]+', ИНН: '+[ITN]+', ОКПО: '+[ARCoEaO] as "Реквизиты фирмы поставщика" from [dbo].[Suppler]  inner join [dbo].[Type\_Supplier] on [Type\_Suppler\_ID] = [ID\_Type\_Supplier]  order by [Name\_Supplier] DESC, [Full\_Name\_Suppler] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Representative]  (  [ID\_Representative] [int] not null identity(1,1),  [Suppler\_ID] [int] not null,  [First\_Name\_Repres] [varchar] (30) not null,  [Second\_Name\_Repres] [varchar] (30) not null,  [Middle\_Name\_Repres] [varchar] (30) null default('-')  constraint [PK\_Representative] primary key clustered  ([ID\_Representative] ASC) on [PRIMARY],  constraint [FK\_Suppler\_Representative] foreign key ([Suppler\_ID])  references [dbo].[Suppler] ([ID\_Suppler])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Representative] ([Suppler\_ID], [First\_Name\_Repres], [Second\_Name\_Repres], [Middle\_Name\_Repres])  values (1,'Романов','Роман','Романович'),  (2,'Кириллов','Кирилл','Кириллович'),  (3,'Владимиров','Владимир','Владимирович')  go |
| Запросы на выборку данных |
| Задача: Вывести в одно поле: тип фирмы, полное название фирмы из объединённых таблиц тип фирмы и фирма, а так же вывести ФИО представителя из объединенной таблицы представитель, отсортировав название типа фирмы по убыванию, а фамилия представителя по возрастанию. |
| select 'Тип организации: '+[Name\_Supplier]+', Полное название: '+[Full\_Name\_Suppler] as "Представляемая фирма", [First\_Name\_Repres]+' '+[Second\_Name\_Repres]+' '+[Middle\_Name\_Repres] as "ФИО представителя" from [dbo].[Representative]  inner join [dbo].[Suppler] on [Suppler\_ID] = [ID\_Suppler]  inner join [dbo].[Type\_Supplier] on [Type\_Suppler\_ID] = [ID\_Type\_Supplier]  order by [Name\_Supplier] DESC, [First\_Name\_Repres] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Treaty]  (  [ID\_Treaty] [int] not null identity(1,1),  [Treaty\_Number] [varchar] (15) not null,  [Urgent] [varchar] (3) not null,  [Term] [varchar] (15) not null,  [Date\_Create\_Term] [date] null default(getdate()),  [Representative\_ID] [int] not null  constraint [PK\_Treaty] primary key clustered  ([ID\_Treaty] ASC) on [PRIMARY],  constraint [UQ\_Treaty\_Number] unique ([Treaty\_Number]),  constraint [CH\_Treaty\_Number] check ([Treaty\_Number] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9]-Д')),  constraint [CH\_Urgent] check ([Urgent] in ('Да','Нет')),  constraint [CH\_Term] check ([Term] like ('[0-9][0-9] лет') or [Term] like ('[0-9][0-9][0-9] месяцев')),  constraint [FK\_Representative\_Treaty] foreign key ([Representative\_ID])  references [dbo].[Representative] ([ID\_Representative])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Treaty] ([Treaty\_Number], [Urgent], [Term], [Representative\_ID])  values ('0000000001-21-Д','Да','05 лет',1),  ('0000000002-21-Д','Да','05 лет',2),  ('0000000003-21-Д','Да','02 лет',3)  go |
| Запросы на выборку данных |
| Задача: Вывести номер договора, объединив с срочностью договора, сроком действия, датой создания договора, информацию об организации и ФИО представителя, отсортировав дату по убыванию и по названию организации по возрастанию. |
| select [Treaty\_Number] as "Номер договора", 'Срочный: '+[Urgent]+'. Срок действия: '+[Term]+'. Дата создания договора: '+CONVERT([varchar](10), [Date\_Create\_Term]) as "Детали договора",  'Тип организации: '+[Name\_Supplier]+', Полное название: '+[Full\_Name\_Suppler]+'. ФИО представителя:'+[First\_Name\_Repres]+' '+[Second\_Name\_Repres]+' '+[Middle\_Name\_Repres] as "Данные организации поставщика"  from [dbo].[Treaty]  inner join [dbo].[Representative] on [Representative\_ID] = [ID\_Representative]  inner join [dbo].[Suppler] on [Suppler\_ID] = [ID\_Suppler]  inner join [dbo].[Type\_Supplier] on [Type\_Suppler\_ID] = [ID\_Type\_Supplier]  order by [Date\_Create\_Term] DESC, [Name\_Supplier] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Buyer\_Card]  (  [ID\_Buyer\_Card] [int] not null identity(1,1),  [Card\_Number] [varchar] (19) not null,  [Card\_Holder] [varchar] (60) not null,  [Validity] [varchar] (5) not null,  [Buyer\_ID] [int] not null  constraint [PK\_Buyer\_Card] primary key clustered  ([ID\_Buyer\_Card] ASC) on [PRIMARY],  constraint [UQ\_Card\_Number] unique ([Card\_Number]),  constraint [CH\_Card\_Number] check ([Card\_Number] like '[0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9]'),  constraint [CH\_Validity] check ([Validity] like ('[0-9][0-9]/[0-9][0-9]')),  constraint [FK\_Buyer\_Buyer\_Card] foreign key ([Buyer\_ID])  references [dbo].[Buyer] ([ID\_Buyer])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Buyer\_Card] ([Card\_Number], [Card\_Holder], [Validity], [Buyer\_ID])  values ('2245 5367 7642 7640','IVANOV IVAN','02/24',1),  ('5542 5674 9974 8974','PETROV PETR','10/25',2),  ('8444 8854 1178 9942','ANDREEV ANDREY','12/24',3)  go |
| Запросы на выборку данных |
| Задача: Вывести ФИО покупателя и реквизиты его банковской карты, отсортировав фамилию по возрастанию. |
| select [First\_Name\_Buyer]+' '+[Second\_Name\_Buyer]+' '+[Middle\_Name\_Buyer] as "Данные покупателя", 'Номер карты: '+[Card\_Number]+'. Держатель: '+[Card\_Holder]+', действительна до: '+[Validity] as "Данные карты" from [dbo].[Buyer]  inner join [dbo].[Buyer\_Card] on [ID\_Buyer] = [Buyer\_ID]  order by [First\_Name\_Buyer] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Combination]  (  [ID\_Combination] [int] not null identity(1,1),  [Post\_ID] [int] not null,  [Employee\_ID] [int] not null  constraint [PK\_Combination] primary key clustered  ([ID\_Combination] ASC) on [PRIMARY],  constraint [FK\_Post\_Combination] foreign key ([Post\_ID])  references [dbo].[Post] ([ID\_Post]),  constraint [FK\_Employee\_Combination] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Combination] ([Post\_ID], [Employee\_ID])  values (1,1),  (2,2),  (1,3)  go |
| Запросы на выборку данных |
| Задача: Вывести ФИО сотрудника, ключ отмены и название соответствующей должности, отсортировать название должности по возрастанию, а фамилию по убыванию. |
| select [First\_Name\_Emp]+' '+[Second\_Name\_Emp]+' '+[Middle\_Name\_Emp] as "ФИО Сотрудника", [Key\_Employee] as "Ключ отмены", [Name\_Post] as "Должность" from [dbo].[Combination]  inner join [dbo].[Employee] on [ID\_Employee] = [Employee\_ID]  inner join [dbo].[Post] on [ID\_Post] = [Post\_ID]  order by [Name\_Post] ASC, [First\_Name\_Emp] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Employee\_Change]  (  [ID\_Employee\_Change] [int] not null identity(1,1),  [Employee\_Change\_Number] [varchar] (11) not null,  [Combination\_ID] [int] not null,  [Cash\_Machine\_ID] [int] not null,  [Date\_Time\_Create\_Emp\_Chg] [datetime] null default(getdate())  constraint [PK\_Employee\_Change] primary key clustered  ([ID\_Employee\_Change] ASC) on [PRIMARY],  constraint [FK\_Combination\_Employee\_Change] foreign key ([Combination\_ID])  references [dbo].[Combination] ([ID\_Combination]),  constraint [FK\_Cash\_Machine\_Employee\_Change] foreign key ([Cash\_Machine\_ID])  references [dbo].[Cash\_Machine] ([ID\_Cash\_Machine])  )  go | alter table [dbo].[Employee\_Change] add constraint [CH\_Employee\_Change\_Number]  check ([Employee\_Change\_Number] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]')  go  alter table [dbo].[Employee\_Change] add constraint [UQ\_Employee\_Change\_Number]  unique ([Employee\_Change\_Number])  go |
| Запросы на добавление данных |
| insert into [dbo].[Employee\_Change] ([Employee\_Change\_Number],[Combination\_ID],[Cash\_Machine\_ID])  values ('00000000001', 1, 1),  ('00000000002', 2, 2),  ('00000000004', 3, 3),  ('00000000005', 2, 2),  ('00000000003', 1, 1)  go |
| Запросы на выборку данных |
| Задача: Вывести ФИО сотрудника и должность в одно поле, номер смены с датой открытия, отсортировав номер смены по убыванию. |
| select 'ФИО: '+[First\_Name\_Emp]+' '+[Second\_Name\_Emp]+' '+[Middle\_Name\_Emp]+', должность: '+[Name\_Post] as "Данные сотрудника",  '№ смены: '+[Employee\_Change\_Number]+', дата открытия: '+convert([varchar] (25), [Date\_Time\_Create\_Emp\_Chg]) as "Данные смены" from  [dbo].[Employee\_Change]  inner join [dbo].[Combination] on [Combination\_ID] = [ID\_Combination]  inner join [dbo].[Employee] on [ID\_Employee] = [Employee\_ID]  inner join [dbo].[Post] on [ID\_Post] = [Post\_ID]  order by [Employee\_Change\_Number] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Product]  (  [ID\_Product] [int] not null identity(1,1),  [Price\_Product] [decimal] (38,2) null default (0.0),  [Product\_Name] [varchar] (50) not null,  [Compound] [varchar] (max) not null,  [End\_Date] [date] not null,  [Vendor\_Code] [varchar] (13) not null,  [Product\_Type\_ID] [int] not null,  [Manufacturer\_ID] [int] not null  constraint [PK\_Product] primary key clustered  ([ID\_Product] ASC) on [PRIMARY],  constraint [UQ\_Vendor\_Code] unique ([Vendor\_Code]),  constraint [CH\_Vendor\_Code] check ([Vendor\_Code] like '[A-Z][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Price\_Product] check ([Price\_Product] >= 0),  constraint [CH\_End\_Date] check ([End\_Date] >= Getdate()),  constraint [FK\_Product\_Type\_Product] foreign key ([Product\_Type\_ID])  references [dbo].[Type\_Product] ([ID\_Type\_Product]),  constraint [FK\_Manufacturer\_Product] foreign key ([Manufacturer\_ID])  references [dbo].[Manufacturer] ([ID\_Manufacturer])  )  go | alter table [dbo].[Product] drop constraint [CH\_End\_Date]  go  alter table [dbo].[Product] alter column [End\_Date] [varchar] (20) not null  go |
| Запросы на добавление данных |
| insert into [dbo].[Product] ([Price\_Product], [Product\_Name], [Compound], [End\_Date],  [Vendor\_Code], [Product\_Type\_ID], [Manufacturer\_ID])  values (340,'Кофе','Кофейные зёрна','72 месяца','F753273-76336',1,1),  (150,'Помидоры','Натуральные вещества','10 дней','R452677-43467',2,2),  (250,'Карбонат','Говядина','14 дней','Y762004-22787',3,3),  (120,'Растительное масло','Взбитое молоко 90%','6 месяцев','U632206-25453',4,4),  (110,'Молоко 5%','Натуральные вещества','12 дней','L653573-54373',4,4)  go |
| Запросы на выборку данных |
| Задача: Вывести название производителя и страну, артикул товара, название продукта с дополнительными полями: цена за одну единицу, вид продукта, описание, срок годности, отсортировав цену продукта по возрастанию. |
| select 'Производитель: '+[Name\_Manufacturer]+', страна: '+[Name\_Country] as "Данные производителя", [Vendor\_Code] as "Артикул",  'Название продукта: '+[Product\_Name]+', цена за шт.: '+CONVERT([varchar] (100), [Price\_Product])+' руб., вид продукта: '+[Name\_Type]+', описание - "'+[Compound]+'", срок годности = '+[End\_Date] as "Данные товара"  from [dbo].[Product]  inner join [dbo].[Type\_Product] on [Product\_Type\_ID] = [ID\_Type\_Product]  inner join [dbo].[Manufacturer] on [ID\_Manufacturer] = [Manufacturer\_ID]  inner join [dbo].[Country] on [ID\_Country] = [Country\_ID]  order by [Price\_Product] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Product\_List\_View]  (  [ID\_Product\_List\_View] [int] not null identity(1,1),  [Product\_List\_ID] [int] not null,  [Product\_ID] [int] not null  constraint [PK\_Product\_List\_View] primary key clustered  ([ID\_Product\_List\_View] ASC) on [PRIMARY],  constraint [FK\_Product\_List\_Product\_List\_View] foreign key ([Product\_List\_ID])  references [dbo].[Product\_List] ([ID\_Product\_List]),  constraint [FK\_Product\_Product\_List\_View] foreign key ([Product\_ID])  references [dbo].[Product] ([ID\_Product])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Product\_List\_View] ([Product\_List\_ID], [Product\_ID])  values (1,1),  (2,2),  (1,3),  (1,4),  (3,5)  go |
| Запросы на выборку данных |
| Задача: Вывести название, артикул и вид продукта, номер сметы и дату создания, отсортировать дату создания сметы по убыванию, артикул товара по возрастанию. |
| select 'Название: '+[Product\_Name]+', артикурл: '+[Vendor\_Code]+', вид товара: '+[Name\_Type] as "Данные товара",  'Номер сметы: '+[Product\_List\_Number]+'. Дата создания - '+convert([varchar] (10),[Date\_Time\_Create\_PL]) as "Данные товарной сметы" from [dbo].[Product\_List\_View]  inner join [dbo].[Product] on [ID\_Product] = [Product\_ID]  inner join [dbo].[Type\_Product] on [Product\_Type\_ID] = [ID\_Type\_Product]  inner join [dbo].[Product\_List] on [ID\_Product\_List] = [Product\_List\_ID]  order by [Date\_Time\_Create\_PL] DESC, [Vendor\_Code] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Cancellation\_View]  (  [ID\_Cancellation\_View] [int] not null identity(1,1),  [Cancellation\_ID] [int] not null,  [Product\_ID] [int] not null  constraint [PK\_Cancellation\_View] primary key clustered  ([ID\_Cancellation\_View] ASC) on [PRIMARY],  constraint [FK\_Cancellation\_Cancellation\_View] foreign key ([Cancellation\_ID])  references [dbo].[Cancellation] ([ID\_Cancellation]),  constraint [FK\_Product\_Cancellation\_View] foreign key ([Product\_ID])  references [dbo].[Product] ([ID\_Product])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Cancellation\_View] ([Cancellation\_ID], [Product\_ID])  values (1,1),  (4,2),  (2,3),  (1,4),  (3,5)  go |
| Запросы на выборку данных |
| Задача: Вывести номер артикула, название продукта, срок годности, стоимость продукта, данные о товаре в одно поле, название производителя, название страны, номер отмены и дату формирования из таблиц состав отмены, продукт, тип продукта, производитель, страна, акт об отмене поставки, отсортировав дату создания акта по убыванию, номер артикула по возрастанию. |
| select 'Артикул: '+[Vendor\_Code]+'. Название - '+[Product\_Name]+', срок годности: '+[End\_Date]+'. Стоимость: '+convert([varchar] (100), [Price\_Product])+' руб.' as "Данные о товаре",  'Производитель: '+[Name\_Manufacturer]+'. Страна: '+[Name\_Country] as "Данные производителя", [Name\_Type] as "Тип товара",  'Номер отмены: '+[Cancellation\_Number]+'. Дата формирования: '+convert([varchar] (10), [Date\_Create\_Cncll])  from [dbo].[Cancellation\_View]  inner join [dbo].[Product] on [Product\_ID] = [ID\_Product]  inner join [dbo].[Type\_Product] on [ID\_Type\_Product] = [Product\_Type\_ID]  inner join [dbo].[Manufacturer] on [ID\_Manufacturer] = [Manufacturer\_ID]  inner join [dbo].[Country] on [ID\_Country] = [Country\_ID]  inner join [dbo].[Cancellation] on [Cancellation\_ID] = [ID\_Cancellation]  order by [Date\_Create\_Cncll] DESC, [Vendor\_Code] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Supply]  (  [ID\_Supply] [int] not null identity(1,1),  [Integrity] [varchar] (30) not null,  [Supply\_Number] [varchar] (15) not null,  [Date\_Create\_Supply] [date] null default (getdate()),  [Time\_Create\_Supply] [time] null default (getdate()),  [Treaty\_ID] [int] not null,  [Warehouse\_ID] [int] not null  constraint [PK\_Supply] primary key clustered  ([ID\_Supply] ASC) on [PRIMARY],  constraint [UQ\_Supply\_Number] unique ([Supply\_Number]),  constraint [CH\_Integrity] check ([Integrity] in ('Нормально','Нарушенное')),  constraint [CH\_Supply\_Number] check ([Supply\_Number] like ('[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9]-П')),  constraint [FK\_Warehouse\_Supply] foreign key ([Warehouse\_ID])  references [dbo].[Warehouse] ([ID\_Warehouse]),  constraint [FK\_Treaty\_Supply] foreign key ([Treaty\_ID])  references [dbo].[Treaty] ([ID\_Treaty])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Supply] ([Integrity], [Supply\_Number], [Treaty\_ID], [Warehouse\_ID])  values ('Нормально','0000000001-21-П',1,1),  ('Нормально','0000000004-21-П',2,2),  ('Нормально','0000000002-21-П',1,1),  ('Нормально','0000000003-21-П',3,2)  go |
| Запросы на выборку данных |
| Задача: Вывести номер поставки, целостность поставки, дату и время поставки, в одном поле, адрес склада, адрес магазина, во второе поле, номер поставки, название поставщика, в третье поле из таблиц: поставка, договор, представитель, поставщик, склад, торговая точка, отсортировав дату создания поставки по убыванию, а время поставки и номер договора по возрастанию. |
| select 'Номер поставки: '+[Supply\_Number]+'. Целостнось: '+[Integrity]+'. Дата и время поставци: '+Convert([varchar](10), [Date\_Create\_Supply])  +' - '+Convert([varchar] (10), [Time\_Create\_Supply]) as "Данные попоставке",  'Адрес склада: '+[Warehouse\_Adress]+'. Адрес магазина: '+[Market\_Address] as "Место доставки",  'Номем: '+[Treaty\_Number]+'. Поставщик: '+[Full\_Name\_Suppler] as "Данные договора" from [dbo].[Supply]  inner join [dbo].[Treaty] on [Treaty\_ID] = [ID\_Treaty]  inner join [dbo].[Representative] on [ID\_Representative] = [Representative\_ID]  inner join [dbo].[Suppler] on [ID\_Suppler] = [Suppler\_ID]  inner join [dbo].[Warehouse] on [ID\_Warehouse] = [Treaty\_ID]  inner join [dbo].[Market\_Place] on [ID\_Market\_Place] = [Market\_Place\_ID]  order by [Date\_Create\_Supply] DESC, [Time\_Create\_Supply] ASC, [Treaty\_Number] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Supply\_View]  (  [ID\_Supply\_View] [int] not null identity(1,1),  [Supply\_Ammount] [int] not null,  [Supply\_ID] [int] not null,  [Product\_ID] [int] not null  constraint [PK\_Supply\_View] primary key clustered  ([ID\_Supply\_View] ASC) on [PRIMARY],  constraint [CH\_Supply\_Ammount] check ([Supply\_Ammount] > 0),  constraint [FK\_Supply\_Supply\_View] foreign key ([Supply\_ID])  references [dbo].[Supply] ([ID\_Supply]),  constraint [FK\_Product\_Supply\_View] foreign key ([Product\_ID])  references [dbo].[Product] ([ID\_Product])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Supply\_View] ([Supply\_Ammount], [Supply\_ID], [Product\_ID])  values (1000, 1, 1),  (1200, 4, 2),  (900, 2, 3),  (2100, 1, 4),  (1750, 3, 5)  go |
| Запросы на выборку данных |
| Задача: Вывести номер поставки, дату и время поставки, в одном поле, адрес склада, торговой точки, во второе поле, название, артикул продукта и количество поставляемого товара в третье поле из таблиц: состав поставки, продукт, поставки, склад, торговая точка, отсортировав дату, время и количество поставляемого товара в поставки по убыванию. |
| select 'Номер поставки: '+[Supply\_Number]+'. Дата и время: '+CONVERT([varchar] (10),[Date\_Create\_Supply])+' - '+CONVERT([varchar] (10), [Time\_Create\_Supply]) as "Общие данные поставки",  'Склад: '+[Warehouse\_Adress]+', торговая точка: '+[Market\_Address] as "Пункт доставки",  'Название продукта: '+[Product\_Name]+', артикул = '+[Vendor\_Code]+', количество в поставке: '+CONVERT([varchar](10),[Supply\_Ammount]) as "Состав доставки" from [dbo].[Supply\_View]  inner join [dbo].[Product] on [ID\_Product] = [Product\_ID]  inner join [dbo].[Supply] on [Supply\_ID] = [ID\_Supply]  inner join [dbo].[Warehouse] on [Warehouse\_ID] = [ID\_Warehouse]  inner join [dbo].[Market\_Place] on [Market\_Place\_ID] = [ID\_Market\_Place]  order by [Date\_Create\_Supply] DESC, [Time\_Create\_Supply] DESC, [Supply\_Ammount] DESC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Sale\_Check]  (  [ID\_Sale\_Check] [int] not null identity(1,1),  [Sale\_Check\_Number] [varchar] (13) not null,  [Date\_Create\_Check] [date] null default(getdate()),  [Time\_Create\_Check] [time] null default(getdate()),  [Deposited\_Ammount] [decimal] (38,2) not null,  [Whith\_VAT] [decimal] (38,2) not null,  [Without\_VAT] [decimal] (38,2) not null,  [Payment\_Type\_ID] [int] not null,  [Employee\_Change\_ID] [int] not null,  [Buyer\_Card\_ID] [int] not null  constraint [PK\_Sale\_Check] primary key clustered  ([ID\_Sale\_Check] ASC) on [PRIMARY],  constraint [UQ\_Sale\_Check\_Number] unique ([Sale\_Check\_Number]),  constraint [CH\_Sale\_Check\_Number] check ([Sale\_Check\_Number] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]-ФК'),  constraint [CH\_Deposited\_Ammount] check ([Deposited\_Ammount] > 0),  constraint [CH\_Whith\_VAT] check ([Whith\_VAT] > 0),  constraint [CH\_Without\_VAT] check ([Without\_VAT] > 0),  constraint [FK\_Payment\_Type\_Sale\_Check] foreign key ([Payment\_Type\_ID])  references [dbo].[Payment\_Type] ([ID\_Payment\_Type]),  constraint [FK\_Employee\_Change\_Sale\_Check] foreign key ([Employee\_Change\_ID])  references [dbo].[Employee\_Change] ([ID\_Employee\_Change]),  constraint [FK\_Buyer\_Card\_Sale\_Check] foreign key ([Buyer\_Card\_ID])  references [dbo].[Buyer\_Card] ([ID\_Buyer\_Card])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Sale\_Check] ([Sale\_Check\_Number], [Deposited\_Ammount], [Whith\_VAT],  [Without\_VAT], [Payment\_Type\_ID], [Employee\_Change\_ID], [Buyer\_Card\_ID])  values ('0000000001-ФК', 900, 804.00, 670.00, 1, 1, 1),  ('0000000002-ФК', 900, 900.00, 750.00, 2, 2, 2),  ('0000000004-ФК', 600, 600.00, 500.00, 1, 4, 3),  ('0000000003-ФК', 144, 144.00, 120.00, 1, 3, 2)  go |
| Запросы на выборку данных |
| Задача: Вывести номер чека, дату и время открытия чека, в одно поле, внесённую сумму с НФС и без НДС, во второе поле, ФИО сотрудника и должность в третье поле, тип оплаты в четвёртое поле из таблиц: товарный чек, карта клиента, смена, совместительство, кассовый аппарат, сотрудник, должность, тип оплаты, отсортировав, номер чека, дату открытия чека по убыванию, время создания по возрастанию. |
| select 'Номер чека -- '+[Sale\_Check\_Number]+'. Дата и время открытия: '+convert([varchar](10), [Date\_Create\_Check])+' '+CONVERT([varchar] (10),[Time\_Create\_Check]) as "Данные чека",  'Внесённая сумма: '+CONVERT([varchar] (10),[Deposited\_Ammount])+' руб. С НДС: '+CONVERT([varchar] (10),[Whith\_VAT])+' руб. Без НДС:'+CONVERT([varchar] (10),[Without\_VAT])+' руб.' as "Оплата",  'ФИО острудника: '+[First\_Name\_Emp]+' '+[Second\_Name\_Emp]+' '+[Middle\_Name\_Emp]+'. Должность: '+[Name\_Post] as "Данные сотрудника",  [Name\_Payment\_Type] as "Тип оплаты" from [dbo].[Sale\_Check]  inner join [dbo].[Buyer\_Card] on [Buyer\_Card\_ID] = [ID\_Buyer\_Card]  inner join [dbo].[Employee\_Change] on [ID\_Employee\_Change] = [Employee\_Change\_ID]  inner join [dbo].[Combination] on [ID\_Combination] = [Combination\_ID]  inner join [dbo].[Cash\_Machine] on [ID\_Cash\_Machine] = [Cash\_Machine\_ID]  inner join [dbo].[Employee] on [ID\_Employee] = [Employee\_ID]  inner join [dbo].[Post] on [ID\_Post] = [Post\_ID]  inner join [dbo].[Payment\_Type] on [ID\_Payment\_Type] = [Payment\_Type\_ID]  order by [Sale\_Check\_Number] desc, [Date\_Create\_Check] DESC, [Time\_Create\_Check] ASC  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Sale\_Check\_View]  (  [ID\_Sale\_Check\_View] [int] not null identity(1,1),  [Sale\_Check\_View\_Ammount] [int] not null,  [Sale\_Check\_ID] [int] not null,  [Product\_ID] [int] not null  constraint [PK\_Sale\_Check\_View] primary key clustered  ([ID\_Sale\_Check\_View] ASC) on [PRIMARY],  constraint [CH\_Sale\_Check\_View\_Ammount] check ([Sale\_Check\_View\_Ammount] > 0),  constraint [FK\_Sale\_Check\_Sale\_Check\_View] foreign key ([Sale\_Check\_ID])  references [dbo].[Sale\_Check] ([ID\_Sale\_Check]),  constraint [FK\_Product\_Sale\_Check\_View] foreign key ([Product\_ID])  references [dbo].[Product] ([ID\_Product])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Sale\_Check\_View] ([Sale\_Check\_View\_Ammount], [Sale\_Check\_ID], [Product\_ID])  values (1, 1, 1),  (5, 2, 2),  (2, 3, 3),  (1, 4, 4),  (3, 1, 5)  go |
| Запросы на выборку данных |
| Задача: Вывести номер чека, дату и время открытия чека, в одно поле, название продукта, цена за 1 единицу, артикул товара количество товара в чеке, в третье поле из таблиц: состав чека, чек, продукт, отсортировав номер чека и дату создания по убыванию. |
| select 'Номер чека -- '+[Sale\_Check\_Number]+'. Дата и время открытия: '+convert([varchar](10), [Date\_Create\_Check])+' '+CONVERT([varchar] (10),[Time\_Create\_Check]) as "Данные чека",  'Название -- '+[Product\_Name]+'; Цена за шт: -- '+CONVERT([varchar] (10),[Price\_Product])+' руб. Артикул = '+[Vendor\_Code]  +'; Кол - во'+CONVERT([varchar] (10),[Sale\_Check\_View\_Ammount]) as "Товар чека"  from [dbo].[Sale\_Check\_View]  inner join [dbo].[Sale\_Check] on [Sale\_Check\_ID] = [ID\_Sale\_Check]  inner join [dbo].[Product] on [ID\_Product] = [Product\_ID]  order by [Sale\_Check\_Number] DESC, [Date\_Create\_Check] DESC  go |
| Результат запроса на выборку данных | |
|  | |

1. Изменение и удаление данных в таблицах

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Type\_Product]  (  [ID\_Type\_Product] [int] not null identity(1,1),  [Name\_Type] [varchar] (50) not null  constraint [PK\_Type\_Product] primary key clustered  ([ID\_Type\_Product] ASC) on [PRIMARY],  constraint [UQ\_Name\_Type] unique ([Name\_Type])  )  go | Изменение данных |
| update [dbo].[Type\_Product] set  [Name\_Type] = 'Овощная продукция'  where  [ID\_Type\_Product] = 2;  go |
|  |
| Удаление данных |
| delete from [dbo].[Type\_Product]  where [ID\_Type\_Product] = 3  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Post]  (  [ID\_Post] [int] not null identity(1,1),  [Name\_Post] varchar (50) not null  constraint [PK\_Post] primary key clustered  ([ID\_Post] ASC) on [PRIMARY],  constraint [UQ\_Name\_Post] unique ([Name\_Post])  )  go | Изменение данных |
| update [dbo].[Post] set  [Name\_Post] = 'Главный продавец'  where  [ID\_Post] = 2  go |
|  |
| Удаление данных |
| delete from [dbo].[Post]  where  [ID\_Post] = 1  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Representative]  (  [ID\_Representative] [int] not null identity(1,1),  [Suppler\_ID] [int] not null,  [First\_Name\_Repres] [varchar] (30) not null,  [Second\_Name\_Repres] [varchar] (30) not null,  [Middle\_Name\_Repres] [varchar] (30) null default('-')  constraint [PK\_Representative] primary key clustered  ([ID\_Representative] ASC) on [PRIMARY],  constraint [FK\_Suppler\_Representative] foreign key ([Suppler\_ID])  references [dbo].[Suppler] ([ID\_Suppler])  )  go | Изменение данных |
| update [dbo].[Representative] set  [Second\_Name\_Repres] = 'Андрей'  where  [ID\_Representative] = 2  go |
|  |
| Удаление данных |
| delete from [dbo].[Representative]  where  [ID\_Representative] = 1  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Buyer\_Card]  (  [ID\_Buyer\_Card] [int] not null identity(1,1),  [Card\_Number] [varchar] (19) not null,  [Card\_Holder] [varchar] (60) not null,  [Validity] [varchar] (5) not null,  [Buyer\_ID] [int] not null  constraint [PK\_Buyer\_Card] primary key clustered  ([ID\_Buyer\_Card] ASC) on [PRIMARY],  constraint [UQ\_Card\_Number] unique ([Card\_Number]),  constraint [CH\_Card\_Number] check ([Card\_Number] like '[0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9]'),  constraint [CH\_Validity] check ([Validity] like ('[0-9][0-9]/[0-9][0-9]')),  constraint [FK\_Buyer\_Buyer\_Card] foreign key ([Buyer\_ID])  references [dbo].[Buyer] ([ID\_Buyer])  )  go | Изменение данных |
| update [dbo].[Buyer\_Card] set  [Card\_Holder] = 'PETROV PAVEL',  [Validity] = '11/25'  where  [ID\_Buyer\_Card] = 2  go |
|  |
| Удаление данных |
| delete from [dbo].[Buyer\_Card]  where  [ID\_Buyer\_Card] = 1  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Sale\_Check\_View]  (  [ID\_Sale\_Check\_View] [int] not null identity(1,1),  [Sale\_Check\_View\_Ammount] [int] not null,  [Sale\_Check\_ID] [int] not null,  [Product\_ID] [int] not null  constraint [PK\_Sale\_Check\_View] primary key clustered  ([ID\_Sale\_Check\_View] ASC) on [PRIMARY],  constraint [CH\_Sale\_Check\_View\_Ammount] check ([Sale\_Check\_View\_Ammount] > 0),  constraint [FK\_Sale\_Check\_Sale\_Check\_View] foreign key ([Sale\_Check\_ID])  references [dbo].[Sale\_Check] ([ID\_Sale\_Check]),  constraint [FK\_Product\_Sale\_Check\_View] foreign key ([Product\_ID])  references [dbo].[Product] ([ID\_Product])  )  go | Изменение данных |
| update [dbo].[Sale\_Check\_View] set  [Product\_ID] = 4,  [Sale\_Check\_ID] = 4  where  [ID\_Sale\_Check\_View] = 5  go |
|  |
| Удаление данных |
| delete from [dbo].[Sale\_Check\_View]  where  [ID\_Sale\_Check\_View] = 4  go |
|  |

1. Фильтрация данных

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Buyer\_Card]  (  [ID\_Buyer\_Card] [int] not null identity(1,1),  [Card\_Number] [varchar] (19) not null,  [Card\_Holder] [varchar] (60) not null,  [Validity] [varchar] (5) not null,  [Buyer\_ID] [int] not null  constraint [PK\_Buyer\_Card] primary key clustered  ([ID\_Buyer\_Card] ASC) on [PRIMARY],  constraint [UQ\_Card\_Number] unique ([Card\_Number]),  constraint [CH\_Card\_Number] check ([Card\_Number] like '[0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9] [0-9][0-9][0-9][0-9]'),  constraint [CH\_Validity] check ([Validity] like ('[0-9][0-9]/[0-9][0-9]')),  constraint [FK\_Buyer\_Buyer\_Card] foreign key ([Buyer\_ID])  references [dbo].[Buyer] ([ID\_Buyer])  )  go | Задача: Вывести данные по карте клиента, где год окончания равен 24. |
| select [Card\_Number], [Card\_Holder] from [dbo].[Buyer\_Card]  where [Validity] like '%24'  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Combination]  (  [ID\_Combination] [int] not null identity(1,1),  [Post\_ID] [int] not null,  [Employee\_ID] [int] not null  constraint [PK\_Combination] primary key clustered  ([ID\_Combination] ASC) on [PRIMARY],  constraint [FK\_Post\_Combination] foreign key ([Post\_ID])  references [dbo].[Post] ([ID\_Post]),  constraint [FK\_Employee\_Combination] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee])  )  go | Задача: Вывести данные о сотрудниках и должностях, где номер должности равен или 2 или 3. |
| select [First\_Name\_Emp]+' '+[Second\_Name\_Emp]+' '+[Middle\_Name\_Emp] as "ФИО Сотрудника", [Key\_Employee] as "Ключ отмены", [Name\_Post] as "Должность" from [dbo].[Combination]  inner join [dbo].[Employee] on [ID\_Employee] = [Employee\_ID]  inner join [dbo].[Post] on [ID\_Post] = [Post\_ID]  where [Post\_ID] in (2,3)  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Product]  (  [ID\_Product] [int] not null identity(1,1),  [Price\_Product] [decimal] (38,2) null default (0.0),  [Product\_Name] [varchar] (50) not null,  [Compound] [varchar] (max) not null,  [End\_Date] [date] not null,  [Vendor\_Code] [varchar] (13) not null,  [Product\_Type\_ID] [int] not null,  [Manufacturer\_ID] [int] not null  constraint [PK\_Product] primary key clustered  ([ID\_Product] ASC) on [PRIMARY],  constraint [UQ\_Vendor\_Code] unique ([Vendor\_Code]),  constraint [CH\_Vendor\_Code] check ([Vendor\_Code] like '[A-Z][0-9][0-9][0-9][0-9][0-9][0-9]-[0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Price\_Product] check ([Price\_Product] >= 0),  constraint [CH\_End\_Date] check ([End\_Date] >= Getdate()),  constraint [FK\_Product\_Type\_Product] foreign key ([Product\_Type\_ID])  references [dbo].[Type\_Product] ([ID\_Type\_Product]),  constraint [FK\_Manufacturer\_Product] foreign key ([Manufacturer\_ID])  references [dbo].[Manufacturer] ([ID\_Manufacturer])  )  go | Задача: Вывести данные о продуктах и типе продукта, где название типа равно «Мясные изделия». |
| select 'Производитель: '+[Name\_Manufacturer]+', страна: '+[Name\_Country] as "Данные производителя", [Vendor\_Code] as "Артикул",  'Название продукта: '+[Product\_Name]+', цена за шт.: '+CONVERT([varchar] (100), [Price\_Product])+' руб., вид продукта: '+[Name\_Type]+', описание - "'+[Compound]+'", срок годности = '+[End\_Date] as "Данные товара"  from [dbo].[Product]  inner join [dbo].[Type\_Product] on [Product\_Type\_ID] = [ID\_Type\_Product]  inner join [dbo].[Manufacturer] on [ID\_Manufacturer] = [Manufacturer\_ID]  inner join [dbo].[Country] on [ID\_Country] = [Country\_ID]  where [Name\_Type] = 'Мясное изделье'  go |
|  |