|  |  |
| --- | --- |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Country] (  [ID\_Country] [int] not null identity(1,1),  [Country\_Name] [varchar] (50) not null  constraint [PK\_Country] primary key clustered  ([ID\_Country] ASC) on [PRIMARY],  constraint [UQ\_Country\_Name] unique ([Country\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Country] ([Country\_Name]) values ('Кувейт'), ('Ливия'), ('Сирия'), ('Сомали'), ('Уганда'), ('Цар'), ('Эритрея'), ('Эфиопия'); |
| Запросы на выборку данных |
| Задача: Вывести название страны, отсортировав название по возрастанию. |
| select [Country\_Name] as 'Название страны' from [dbo].[Country] order by [Country\_Name] desc go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Manufacturer] (  [ID\_Manufacturer] [int] not null identity(1,1),  [Manufacturer\_Name] [varchar] (50) not null,  [Manufacturer\_Country\_ID] [int] not null  constraint [PK\_Manufacturer] primary key clustered  ([ID\_Manufacturer] ASC) on [PRIMARY],  constraint [UQ\_Manufacturer\_Name] unique ([Manufacturer\_Name]),  constraint [FK\_Country\_Manufacturer] foreign key ([Manufacturer\_Country\_ID])  references [dbo].[Country] ([ID\_Country]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Manufacturer] ([Manufacturer\_Name], [Manufacturer\_Country\_ID]) values ('Nokia', 1), ('Vagner', 2), ('Kia', 3), ('Lada', 4),('Bugatti', 2), ('РУССКИЕ ТРАВЫ', 2), ('ИГИЛ', 1); |
| Запросы на выборку данных |
| Задача: Вывести названия производителя и его страны, отсортировав название по возрастанию. |
| select [Manufacturer].[Manufacturer\_Name] as 'Название производителя',  [Country].[Country\_Name] as 'Страна производителя' from Manufacturer  join Country on Country.ID\_Country = Manufacturer.Country\_ID order by [Manufacturer\_Name] |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Buyer] (  [ID\_Buyer] [int] not null identity(1,1),  [Buyer\_Login] [varchar] (32) not null,  [Buyer\_Password] [varchar] (32) not null,  [Buyer\_Phone\_Number] [varchar] (17) not null,  [Buyer\_Address] [varchar] (max) not null,  constraint [PK\_Buyer] primary key clustered  ([ID\_Buyer] ASC) on [PRIMARY],  constraint [UQ\_Buyer\_Login] unique ([Buyer\_Login]),  constraint [UQ\_Buyer\_Phone\_Number] unique ([Buyer\_Phone\_Number]),  constraint [CH\_Buyer\_Login] check (*len*([Buyer\_Login])>=8),  constraint [CH\_Buyer\_Password\_Upper] check ([Buyer\_Password] like ('%[A-Z]%')),  constraint [CH\_Buyer\_Password\_Lower] check ([Buyer\_Password] like ('%[a-z]%')),  constraint [CH\_Buyer\_Password\_Symbols] check ([Buyer\_Password] like ('%[!@#$%^&\*()]%')),  constraint [CH\_Buyer\_Phone\_Number] check ([Buyer\_Phone\_Number] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Buyer] (Buyer\_Login, Buyer\_Password, Buyer\_Phone\_Number, Buyer\_Address) values ('Rootroot', 'R@0t', '+7(777)777-77-77', '798595, Владимирская область, город Подольск, пл. Славы, 23'),  ('Userbebra1', 'R@0t', '+7(111)111-11-11', '023316, Ленинградская область, город Озёры, пер. Сталина, 75'),  ('Userbebra2', 'R@0t', '+7(222)222-22-22', '155667, Смоленская область, город Егорьевск, въезд Космонавтов, 44'),  ('Userbebra3', 'R@0t', '+7(333)333-33-33', '584271, Ивановская область, город Ступино, наб. Ладыгина, 11'); |
| Запросы на выборку данных |
| Задача: Вывести данные покупателей, отсортировав логины по возрастанию. |
| select [Buyer\_Login] as 'Логин покупателя', [Buyer\_Password] as 'Пароль покупателя',   [Buyer\_Phone\_Number] as 'Телефонный номер покупателя', [Buyer\_Address] as 'Адрес покупателя'  from [dbo].[Buyer] order by [Buyer\_Login] desc |
| 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,  [Card\_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\_Card\_Holder] check ([Card\_Holder] = *UPPER*([Card\_Holder])),  constraint [CH\_Card\_Validity] check ([Card\_Validity] like ('[0-1][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], [Card\_Validity], [Buyer\_ID]) values ('1111 1111 1111 1111', 'FIRST BUYER', '01/01', 2),  ('2222 2222 2222 2222', 'SECOND BUYER', '02/02', 3),  ('9999 9999 9999 9999', 'FIRST BUYER', '09/09', 4) |
| Запросы на выборку данных |
| Задача: Вывести данные карт покупателей, отсортировав логины по убыванию. |
| select [Card\_Number] as 'Номер карты', [Card\_Holder] as 'Держатель карты',   [Card\_Validity] as 'Срок истечения карты', [Buyer\_Login] as 'Логин покупателя'  from Buyer\_Card join Buyer B on B.ID\_Buyer = Buyer\_Card.Buyer\_ID order by [Buyer\_Login] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Supplier] (  [ID\_Supplier] [int] not null identity(1,1),  [Supplier\_Full\_Name] [varchar] (max) not null,  [Supplier\_Short\_Name] [varchar] (50) not null,  [Supplier\_Legal\_Address] [varchar] (max) not null,  [Supplier\_Physical\_Address] [varchar] (max) not null,  [Supplier\_E\_Mail] [varchar] (max) not null,  [Supplier\_Phone\_Number] [varchar] (17) not null  constraint [PK\_Supplier] primary key clustered  ([ID\_Supplier] ASC) on [PRIMARY],  constraint [UQ\_Supplier\_Short\_Name] unique ([Supplier\_Short\_Name]),  constraint [UQ\_Supplier\_Phone\_Number] unique ([Supplier\_Phone\_Number]),  constraint [CH\_Supplier\_E\_Mail] check ([Supplier\_E\_Mail] like ('%@%.%')),  constraint [CH\_Supplier\_Phone\_Number] check ([Supplier\_Phone\_Number] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]') ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Supplier] ([Supplier\_Full\_Name], [Supplier\_Short\_Name],[Supplier\_Legal\_Address], [Supplier\_Physical\_Address],[Supplier\_E\_Mail], [Supplier\_Phone\_Number]) values ('BuerakCongressman.USACrackBundle', 'CUM', 'USA, Tokio, Снежая улица', 'Подворотня', 'really@unreal.email', '+7(123)456-78-91'),  ('Navy\_Aluminium0Shih\_Tzu', 'UWU', 'Москва, Воронежская переулок 12 Э', 'Кабинет физики', 'first@second.third', '+7(011)121-31-41'),  ('Olive\_Flea\_Sonata', 'LOLi', 'Пакистан, 7755', 'Галакика Андромеда', 'wtfis@this.email', '+7(516)171-81-91'); |
| Запросы на выборку данных |
| Задача: Вывести данные поставщиков, отсортировав краткие названия по убыванию. |
| select [Supplier\_Full\_Name] as 'Полное название поставщика', [Supplier\_Short\_Name] as 'Краткое название поставщика',  [Supplier\_Legal\_Address] as 'Легальный адрес поставщика', [Supplier\_Physical\_Address] as 'Физический адрес поставщика',  [Supplier\_E\_Mail] as 'Электронная почта поставщика', [Supplier\_Phone\_Number] as 'Телефон поставщика' from [dbo].[Supplier] order by [Supplier\_Short\_Name] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Representative] (  [ID\_Representative] [int] not null identity(1,1),  [Supplier\_ID] [int] not null,  [Repres\_First\_Name] [varchar] (30) not null,  [Repres\_Second\_Name] [varchar] (30) not null,  [Repres\_Middle\_Name] [varchar] (30) null default('-')  constraint [PK\_Representative] primary key clustered  ([ID\_Representative] ASC) on [PRIMARY],  constraint [FK\_Supplier\_Representative] foreign key ([Supplier\_ID])  references [dbo].[Supplier] ([ID\_Supplier]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Representative] ([Supplier\_ID], [Repres\_First\_Name], [Repres\_Second\_Name], [Repres\_Middle\_Name]) values (1, 'Howard', 'Kunze', 'Bar'),  (2, 'Billy', 'Casper', 'BTS'),  (3, 'Elvis', 'Grant', 'UAR'),  (1, 'Catharine', 'Becker', 'NAN'); |
| Запросы на выборку данных |
| Задача: Вывести ФИО представителей поставщиков и краткие названия, отсортировав названия по убыванию. |
| select [Supplier\_Short\_Name] as 'Поставщик', ([Repres\_Second\_Name] + ' ' + [Repres\_First\_Name] + ' ' + [Repres\_Middle\_Name]) as 'ФИО представителя' from [dbo].[Representative] join Supplier on Supplier.ID\_Supplier = Representative.Supplier\_ID order by [Supplier\_Short\_Name] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Contract] (  [ID\_Contract] [int] not null identity(1,1),  [Representative\_ID] [int] not null,  [Contract\_Expiry\_Date] [date] not null,  [Contract\_Conclusion\_Date] [date] null default(*getdate*())  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [CH\_Contract\_Expiry\_Date] check ([Contract\_Expiry\_Date] >= *CAST*(*getdate*() as date)),  constraint [CH\_Contract\_Conclusion\_Date] check ([Contract\_Conclusion\_Date] <= *CAST*(*getdate*() as date)),  constraint [FK\_Representative\_Contract] foreign key ([Representative\_ID])  references [dbo].[Representative] ([ID\_Representative]) ) go |  |
| Запросы на добавление данных |
| insert into [dbo].[Contract] ([Representative\_ID], [Contract\_Expiry\_Date]) values (1, *DATEADD*(month, 1, *getdate*())),(2, *DATEADD*(year, 1, *getdate*())),(4, *DATEADD*(year, 5, *getdate*())); |
| Запросы на выборку данных |
| Задача: Вывести фио представителя поставщика и срок действия договора отсортировав по дате заключения |
| select ([Repres\_Second\_Name] + ' ' + [Repres\_First\_Name] + ' ' + [Repres\_Middle\_Name]) as 'ФИО представителя поставщика', *CONVERT*(varchar, [Contract\_Conclusion\_Date]) + '-' + *CONVERT*(varchar, [Contract\_Expiry\_Date]) as 'Сроки действия договора' from [dbo].[Contract] join Representative on Representative.ID\_Representative = Contract.Representative\_ID order by [Contract\_Conclusion\_Date] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Characteristics\_Name] (  [ID\_Characteristic\_Name] [int] not null identity(1,1),  [Characteristic\_Name] [varchar] (30) not null  constraint [PK\_Characteristic\_Name] primary key clustered  ([ID\_Characteristic\_Name] ASC) on [PRIMARY],  constraint [UQ\_Characteristic\_Name] unique ([Characteristic\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into dbo.Characteristics\_Name (Characteristic\_Name) values ('Профиль XMP'),  ('Тип памяти'),  ('Частота DDR'),  ('Частота ГП'),  ('Частота ЦП'); |
| Запросы на выборку данных |
| Задача: Вывести названия характеристик товаров, отсортировав названия по возрастанию. |
| select [Characteristic\_Name] as 'Название характеристия' from [dbo].[Characteristics\_Name] order by [Characteristic\_Name] asc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Goods\_Type] (  [ID\_Goods\_Type] [int] not null identity(1,1),  [Goods\_Type\_Name] [varchar] (30) not null  constraint [PK\_Goods\_Type] primary key clustered  ([ID\_Goods\_Type] ASC) on [PRIMARY],  constraint [UQ\_Goods\_Type\_Name] unique ([Goods\_Type\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into dbo.Goods\_Type ([Goods\_Type\_Name]) values ('Блоки питания'),  ('Графические ускорители'),  ('Корпуса ПК'),  ('Оперативная память'),  ('Процессоры'); |
| Запросы на выборку данных |
| Задача: Вывести типы товаров, отсортировав названия по возрастанию. |
| select [Goods\_Type\_Name] as 'Название типа товара' from [dbo].[Goods\_Type] order by [Goods\_Type\_Name] asc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Outlets\_Name] (  [ID\_Name] [int] not null identity(1,1),  [Outlet\_Name] [varchar] (50) not null  constraint [PK\_Outlets\_Name] primary key clustered  ([ID\_Name] ASC) on [PRIMARY],  constraint [UQ\_Outlet\_Name] unique ([Outlet\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Outlets\_Name] ([Outlet\_Name]) values ('DNS'),  ('Pleer.ru'),  ('М.Видео'),  ('Регард'); |
| Запросы на выборку данных |
| Задача: Вывести названия торговых точек, отсортировать по возрастанию. |
| select [Outlet\_Name] as 'Название торговой точки' from [dbo].[Outlets\_Name] order by [Outlet\_Name] asc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Outlet] (  [ID\_Outlet] [int] not null identity(1,1),  [Outlet\_Adress] [varchar] (MAX) not null,  [Name\_ID] [int] not null  constraint [PK\_Outlet] primary key clustered  ([ID\_Outlet] ASC) on [PRIMARY],  constraint [FK\_Name\_Outlet] foreign key ([Name\_ID])  references [dbo].[Outlets\_Name] ([ID\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Outlet] ([Outlet\_Adress], [Name\_ID]) values ('063447, Тюменская область, город Наро-Фоминск, спуск Домодедовская, 68', 1),  ('099001, Брянская область, город Озёры, пл. Ленина, 79', 2),  ('508768, Тамбовская область, город Дорохово, наб. Сталина, 51', 3),  ('522727, Рязанская область, город Лотошино, пл. Косиора, 52', 3),  ('901387, Брянская область, город Подольск, ул. Ленина, 96', 3); |
| Запросы на выборку данных |
| Задача: Вывести адреса и названия торговых точек, отсортировав названия по djphfcnfyb.. |
| select [Outlet\_Adress] as 'Адрес торговой точки', [Outlet\_Name] as 'Название торговой точки' from [dbo].[Outlet] join Outlets\_Name on Outlets\_Name.ID\_Name = Outlet.Name\_ID order by [Outlet\_Name] asc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Post] (  [ID\_Post] [int] not null identity(1,1),  [Post\_Name] [varchar] (50) not null,  [Superior\_Post\_ID] [int] not null,  [Post\_Salary] [int] not null  constraint [PK\_Post] primary key clustered  ([ID\_Post] ASC) on [PRIMARY],  constraint [UQ\_Post\_Name] unique ([Post\_Name]),  constraint [CH\_Post\_Salary] check ([Post\_Salary] > 0),  constraint [FK\_Superior\_Post] foreign key ([Superior\_Post\_ID])  references [dbo].[Post] ([ID\_Post]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Post] ([Post\_Name], [Superior\_Post\_ID], [Post\_Salary]) values ('Директор', 1, 500000),  ('Зам.Директора', 1, 300000),  ('Управляющий Филиалом', 2, 150000),  ('Главный Техник Филлиала', 3, 100000),  ('Кассир', 3, 50000),  ('Уборщик', 3, 30000),  ('Помощник Техника', 4, 15000); |
| Запросы на выборку данных |
| Задача: Вывести имя должности, превосходящей её и зарплату, отсортировав зарплаты по убыванию. |
| select P.[Post\_Name] as 'Имя должности' ,  S.[Post\_Name] as 'Имя превосходящей должности' ,  P.[Post\_Salary] as 'Зарплата' from [dbo].[Post] P  join [dbo].[Post] S on S.ID\_Post = P.Superior\_Post\_ID order by P.[Post\_Salary] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Employee] (  [ID\_Employee] [int] not null identity(1,1),  [Emp\_First\_Name] [varchar] (30) not null,  [Emp\_Second\_Name] [varchar] (30) not null,  [Emp\_Middle\_Name] [varchar] (30) null default ('-'),  [Post\_ID] [int] not null,  [Outlet\_ID] [int] not null,  [Emp\_Login] [varchar] (32) not null,  [Emp\_Password] [varchar] (32) not null  constraint [PK\_Employee] primary key clustered  ([ID\_Employee] ASC) on [PRIMARY],  constraint [UQ\_Emp\_Login] unique ([Emp\_Login]),  constraint [CH\_Emp\_Login] check (*len*([Emp\_Login])>=8),  constraint [CH\_Emp\_Password\_Upper] check ([Emp\_Password] like ('%[A-Z]%')),  constraint [CH\_Emp\_Password\_Lower] check ([Emp\_Password] like ('%[a-z]%')),  constraint [CH\_Emp\_Password\_Symbols] check ([Emp\_Password] like ('%[!@#$%^&\*()]%')),  constraint [FK\_Post\_Emp] foreign key ([Post\_ID])  references [dbo].[Post] ([ID\_Post]),  constraint [FK\_Outlet\_Emp] foreign key ([Outlet\_ID])  references [dbo].[Outlet] ([ID\_Outlet]) ) go | - |
| Запросы на добавление данных |
| insert into dbo.Employee ([Emp\_First\_Name], [Emp\_Second\_Name], [Emp\_Middle\_Name], [Post\_ID], [Outlet\_ID], [Emp\_Login], [Emp\_Password]) values ('Root', 'Root', 'Root', 1, 1, 'Rootroot', 'R@0t'),  ('Мария', 'Чижова', 'Мироновна', 4, 1, 'Outlet1\_User4', 'R@0t'),  ('Алина', 'Петрова', 'Сергеевна', 5, 1, 'Outlet1\_User5', 'R@0t'),  ('Руслан', 'Михайлов', 'Дмитриевич', 4, 2, 'Outlet2\_User4', 'R@0t'),  ('Александра', 'Самойлова', 'Максимовна', 5, 2, 'Outlet2\_User5', 'R@0t'); |
| Запросы на выборку данных |
| Задача: Вывести ФИО сотрудников, их должность, логин, пароль и название торговой точки, отсортировав по адресу торговой точки |
| select ([Emp\_First\_Name] + ' ' + [Emp\_Second\_Name] + ' ' + [Emp\_Middle\_Name]) as 'ФИО Сотрудника',  [Post\_Name] as 'Должность',  [Outlet\_Name] as 'Название торговой точки',  [Emp\_Login] as 'Логин сотрудника',  [Emp\_Password] as 'Пароль сотрудника' from [dbo].[Employee] join Post P on P.ID\_Post = Employee.Post\_ID join Outlet O on O.ID\_Outlet = Employee.Outlet\_ID join Outlets\_Name N on N.ID\_Name = O.Name\_ID order by [Outlet\_Adress] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Passport] (  [ID\_Passport] [int] not null identity(1,1),  [Buyer\_ID] [int] not null,  [Buyer\_First\_Name] [varchar] (30) not null,  [Buyer\_Second\_Name] [varchar] (30) not null,  [Buyer\_Middle\_Name] [varchar] (30) null default ('-'),  [Passport\_Series] [varchar] (5) not null,  [Passport\_Code] [varchar] (6) not null,  [Buyer\_Gender] [varchar] (3) not null,  [Buyer\_Birthday] [date] not null,  [Buyer\_Birthplace] [varchar] (30) not null,  [Passport\_Issued\_By] [varchar] (100) not null  constraint [PK\_Passport] primary key clustered  ([ID\_Passport] ASC) on [PRIMARY],  constraint [FK\_Passport\_Buyer] foreign key ([Buyer\_ID])  references [dbo].[Buyer] ([ID\_Buyer]),  constraint [CH\_Passport\_Series] check ([Passport\_Series] like ('[0-9][0-9] [0-9][0-9]')),  constraint [UQ\_Passport\_Code] unique ([Passport\_Code]),  constraint [CH\_Passport\_Code] check ([Passport\_Code] like '[0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Buyer\_Gender] check ([Buyer\_Gender] in ('МУЖ','ЖЕН')),  constraint [CH\_Buyer\_Birthday] check (*DATEDIFF*(YEAR, [Buyer\_Birthday], *getdate*()) >= 14) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Passport] ([Buyer\_ID], [Buyer\_First\_Name], [Buyer\_Second\_Name], [Buyer\_Middle\_Name], [Passport\_Series], [Passport\_Code], [Buyer\_Gender], [Buyer\_Birthday], [Buyer\_Birthplace], [Passport\_Issued\_By]) values (2, 'Рут', 'Рутов', 'Рутович', '22 22', '222222', 'МУЖ', '2002-02-02', 'ГОР. ПОДОЛЬСК', 'ОТДЕЛОМ ВНУТРЕННИХ ДЕЛ НЕСУЩЕСТВУЮЩЕГО РАЙОНА Г.ПОДОЛЬСК'),  (3, 'Ольга', 'Акимова', 'Михайловна', '33 33', '333333', 'ЖЕН', '2003-03-03', 'ГОР. ОЗЁРЫ', 'ОТДЕЛОМ ВНУТРЕННИХ ДЕЛ НЕСУЩЕСТВУЮЩЕГО РАЙОНА Г.ОЗЁРЫ'),  (4, 'Алексей', 'Сальников', 'Даниилович', '44 44', '444444', 'МУЖ', '2004-04-04', 'Г. ЕГОРЬЕВСК', 'ОТДЕЛОМ ВНУТРЕННИХ ДЕЛ НЕСУЩЕСТВУЮЩЕГО РАЙОНА Г.ЕГОРЬЕВСК'); |
| Запросы на выборку данных |
| Задача: Вывести Логин покупателя, ФИО, серию с номером паспорта, пол, дату и место рождения и кто выдал паспорт отсортировав по логину. |
| select [Buyer\_Login] as 'Логин покупателя',  ([Buyer\_First\_Name] + ' ' + [Buyer\_Second\_Name] + ' ' + [Buyer\_Middle\_Name])  as 'ФИО',  ([Passport\_Series] + ' ' + [Passport\_Code]) as 'Серия и номер паспорта',  [Buyer\_Gender] as 'Пол',  [Buyer\_Birthday] as 'Дата рождения',  [Buyer\_Birthplace] as 'Место рождения',  [Passport\_Issued\_By] as 'Паспорт выдан' from [dbo].[Passport] join Buyer B on B.ID\_Buyer = Passport.Buyer\_ID order by [Buyer\_Login] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Payment\_Type] (  [ID\_Payment\_Type] [int] not null identity(1,1),  [Payment\_Type\_Name] [varchar] (30) not null  constraint [PK\_Payment\_Type] primary key clustered  ([ID\_Payment\_Type] ASC) on [PRIMARY],  constraint [UQ\_Payment\_Type] unique ([Payment\_Type\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Payment\_Type] ([Payment\_Type\_Name]) values ('Картой курьеру'),('Наличными курьеру'),('Переводом курьеру'); |
| Запросы на выборку данных |
| Задача: Вывести названия типов оплаты отсортировав по возрастанию. |
| select [Payment\_Type\_Name] as 'Название типа оплаты' from [Payment\_Type] order by [Payment\_Type\_Name] asc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Order] (  [ID\_Order] [int] not null identity(1,1),  [Buyer\_Card\_ID] [int] not null,  [Order\_Date] [date] null default *getdate*(),  [Order\_Time] [time] null default *getdate*(),  [Payment\_Type\_ID] [int] not null  constraint [PK\_Order] primary key clustered  ([ID\_Order] ASC) on [PRIMARY],  constraint [FK\_Card\_Order] foreign key ([Buyer\_Card\_ID])  references [dbo].[Buyer\_Card] ([ID\_Buyer\_Card]),  constraint [CH\_Order\_Date] check ([Order\_Date] = *cast*(*getdate*() as date)),  constraint [CH\_Order\_Time] check ([Order\_Time] between *cast*(*getdate*() as time) and *DATEADD*(MINUTE, 1, *cast*(*getdate*() as time))),  constraint [FK\_Payment\_Type\_Order] foreign key ([Payment\_Type\_ID])  references [dbo].[Payment\_Type] ([ID\_Payment\_Type]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Order] ([Buyer\_Card\_ID], [Payment\_Type\_ID]) values (1, 2), (2, 3); |
| Запросы на выборку данных |
| Задача: Вывести ФИО покупателя, дату и время заказа и тип оплаты, отсортировав по дате и времени заказа. |
| select ([Buyer\_First\_Name] + ' ' + [Buyer\_Second\_Name] + ' ' + [Buyer\_Middle\_Name]) as 'ФИО покупателя',  *CONCAT*([Order\_Date], ' ', *DATEPART*(hour, [Order\_Time]), ':', *DATEPART*(minute , [Order\_Time])) as 'Дата и время заказа',  [Payment\_Type\_Name] from [dbo].[Order]  join Buyer\_Card BC on BC.ID\_Buyer\_Card = [Order].Buyer\_Card\_ID  join Buyer B on B.ID\_Buyer = BC.Buyer\_ID  join Passport P on BC.Buyer\_ID = P.Buyer\_ID  join Payment\_Type PT on PT.ID\_Payment\_Type = [Order].Payment\_Type\_ID order by [Order\_Date], [Order\_Time] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Receipt] (  [ID\_Receipt] [int] not null identity(1,1),  [Order\_ID] [int] not null,  [Receipt\_Sum] [int] not null,  [Receipt\_Total] [int] not null,  [Employee\_ID] [int] not null,  [Closure\_Date] [date] null default *getdate*(),  [Closure\_Time] [time] null default *getdate*()  constraint [PK\_Receipt] primary key clustered  ([ID\_Receipt] ASC) on [PRIMARY],  constraint [FK\_Order\_Receipt] foreign key ([Order\_ID])  references [dbo].[Order] ([ID\_Order]),  constraint [CH\_Receipt\_Sum] check ([Receipt\_Sum] > 0),  constraint [CH\_Receipt\_Total] check ([Receipt\_Total] >= 0),  constraint [FK\_Employee\_Receipt] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [CH\_Closure\_Date] check ([Closure\_Date] = *cast*(*getdate*() as date)),  constraint [CH\_Closure\_Time] check ([Closure\_Time] between *cast*(*getdate*() as time) and *DATEADD*(MINUTE, 1, *cast*(*getdate*() as time))) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Receipt] ([Order\_ID], [Receipt\_Sum], [Receipt\_Total], [Employee\_ID]) values (12, 10000, 10000, 3); |
| Запросы на выборку данных |
| Задача: Вывести сумму заказа, итого, ФИО сотрудника и дату с временем закрытия заказа, отсортировав по дате и времени. |
| select Receipt\_Sum as 'Сумма заказа',  Receipt\_Total as 'Итого', ([Emp\_First\_Name] + ' ' + [Emp\_Second\_Name] + ' ' + [Emp\_Middle\_Name]) as 'ФИО Сотрудника',  *CONCAT*(Closure\_Date, ' ', *DATEPART*(hour, Closure\_Time), ':', *DATEPART*(minute, Closure\_Time)) as 'Дата и время закрытия заказа' from [dbo].[Receipt] join Employee E on E.ID\_Employee = Receipt.Employee\_ID order by [Closure\_Date], [Closure\_Time] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Requests\_For\_Supply] (  [ID\_Request] [int] not null identity(1,1),  [Request\_Status] [varchar] (10) not null,  [Request\_Date] [date] null default *getdate*(),  [Request\_Time] [time] null default *getdate*(),  [Contract\_ID] [int] not null,  [Employee\_ID] [int] not null  constraint [PK\_Requests\_For\_Supply] primary key clustered  ([ID\_Request] ASC) on [PRIMARY],  constraint [CH\_Request\_Status] check ([Request\_Status] in ('ОТПРАВЛЕНО','ПОЛУЧЕНО','НАРУШЕНО')),  constraint [CH\_Request\_Date] check ([Request\_Date] = *cast*(*getdate*() as date)),  constraint [CH\_Request\_Time] check ([Request\_Time] between *cast*(*getdate*() as time) and *DATEADD*(MINUTE, 1, *cast*(*getdate*() as time))),  constraint [FK\_Contract\_Request] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract]),  constraint [FK\_Employee\_Request] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Requests\_For\_Supply] (Request\_Status, Contract\_ID, Employee\_ID) values ('ОТПРАВЛЕНО', 3, 4), ('ПОЛУЧЕНО', 3, 2); |
| Запросы на выборку данных |
| Задача: Вывести статус, дату и время поставки, имя поставщика и фио сотрудника отсортировав по дате и времени. |
| select [Request\_Status] as 'Статус поставки',  *CONCAT*([Request\_Date], ' ', *DATEPART*(hour, [Request\_Time]), ':',  *DATEPART*(minute, [Request\_Time])) as 'Дата и время поставки',  [Supplier\_Short\_Name] as 'Имя поставщика',  ([Emp\_First\_Name] + ' ' + [Emp\_Second\_Name] + ' ' + [Emp\_Middle\_Name]) as 'ФИО Сотрудника' from [dbo].[Requests\_For\_Supply] join Contract C on C.ID\_Contract = Requests\_For\_Supply.Contract\_ID join Representative R2 on R2.ID\_Representative = C.Representative\_ID join Supplier S on S.ID\_Supplier = R2.Supplier\_ID join Employee E on Requests\_For\_Supply.Employee\_ID = E.ID\_Employee order by [Request\_Date], [Request\_Time] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Goods] (  [ID\_Goods] [int] not null identity(1,1),  [Goods\_Name] [varchar] (100) not null,  [Goods\_Article] [varchar] (100) not null,  [Goods\_ID] [int] not null,  [Manufacturer\_ID] [int] not null,  [Goods\_Price] [decimal] (10,2) not null,  [Photo\_Tag] [int] not null,  [Goods\_Type\_ID] [int] not null  constraint [PK\_Goods] primary key clustered  ([ID\_Goods] ASC) on [PRIMARY],  constraint [UQ\_Goods\_Article] unique ([Goods\_Article]),  constraint [FK\_Goods\_Goods] foreign key ([Goods\_ID])  references [dbo].[Goods] ([ID\_Goods]),  constraint [FK\_Manufacturer\_Goods] foreign key ([Manufacturer\_ID])  references [dbo].[Manufacturer] ([ID\_Manufacturer]),  constraint [CH\_Goods\_Price] check ([Goods\_Price] > 0),  constraint [UQ\_Photo\_Tag] unique ([Photo\_Tag]),  constraint [FK\_Type\_Goods] foreign key ([Goods\_Type\_ID])  references [dbo].[Goods\_Type] ([ID\_Goods\_Type]) ) go |  |
| Запросы на добавление данных |
| insert into [dbo].[Goods] (Goods\_Name, Goods\_Article, Goods\_ID, Manufacturer\_ID, Goods\_Price, Photo\_Tag, Goods\_Type\_ID) values ('Видеокарта GIGABYTE GeForce RTX 4090 GAMING OC', 'GV-N4090GAMING OC-24GD', 1, 5, 170000.00, '0B4A38E366B06A267A2389880A428B526D8011E88C5B29AC5FDCE13849B7E2FD', 2); go |
| Запросы на выборку данных |
| Задача: Вывести название с артикулом товара, комплектность, производителя, цену и тег фотографий отсортировав по цене. |
| select *CONCAT*(Goods\_Name, ' ', Goods\_Article) as 'Товар',  CASE [Goods\_ID]  when [ID\_Goods] then 'Нет'  else (select M.Photo\_Tag from [dbo].[Goods] M where M.ID\_Goods = G.Goods\_ID)  end as 'В комплекте с',  [Manufacturer\_Name] as 'Производитель',  [Goods\_Price] as 'Цена',  [Photo\_Tag] as 'Тег фотографий товара',  [Goods\_Type\_Name] as 'Тип товара' from [dbo].[Goods] G  join Manufacturer M on M.ID\_Manufacturer = G.Manufacturer\_ID  join Goods\_Type GT on GT.ID\_Goods\_Type = G.Goods\_Type\_ID order by [Goods\_Price] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Order\_Parts] (  [ID\_Order\_Parts] [int] not null identity(1,1),  [Goods\_ID] [int] not null,  [Order\_Goods\_Amount] [int] not null,  [Order\_ID] [int] not null  constraint [PK\_Order\_Parts] primary key clustered  ([ID\_Order\_Parts] ASC) on [PRIMARY],  constraint [FK\_Order\_Parts\_Goods] foreign key ([Goods\_ID])  references [dbo].[Goods] ([ID\_Goods]),  constraint [CH\_Order\_Goods\_Amount] check ([Order\_Goods\_Amount] > 0),  constraint [FK\_Order\_Parts\_Order] foreign key ([Order\_ID])  references [dbo].[Order] ([ID\_Order]) ) go |  |
| Запросы на добавление данных |
| insert into [dbo].[Order\_Parts] (Goods\_ID, Order\_Goods\_Amount, Order\_ID) values (1, 1, 1); |
| 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 *CONCAT*(Goods\_Name, ' ', Goods\_Article) as 'Товар', [Order\_Goods\_Amount] as 'Количество' from [dbo].[Order\_Parts] join Goods G on G.ID\_Goods = Order\_Parts.Goods\_ID |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Requests\_Parts] (  [ID\_Requests\_Parts] [int] not null identity(1,1),  [Goods\_ID] [int] not null,  [Request\_ID] [int] not null,  [Requests\_Goods\_Amount] [int] not null  constraint [PK\_Requests\_Parts] primary key clustered  ([ID\_Requests\_Parts] ASC) on [PRIMARY],  constraint [FK\_Requests\_Parts\_Goods] foreign key ([Goods\_ID])  references [dbo].[Goods] ([ID\_Goods]),  constraint [FK\_Requests\_Parts\_Request] foreign key ([Request\_ID])  references [dbo].[Requests\_For\_Supply] ([ID\_Request]),  constraint [CH\_Requests\_Goods\_Amount] check ([Requests\_Goods\_Amount] > 0) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Requests\_Parts] ([Goods\_ID], [Request\_ID], [Requests\_Goods\_Amount]) values (1, 2, 10); |
| Запросы на выборку данных |
| Задача: Вывести название товара, статус поставки и количество отсортировав по дате поставки. |
| select [Goods\_Name] as 'Товар', [Request\_Status] as 'Статус',  [Requests\_Goods\_Amount] as 'Количесство' from [dbo].[Requests\_Parts]  join Requests\_For\_Supply RFS on RFS.ID\_Request = Requests\_Parts.Request\_ID  join Goods G on G.ID\_Goods = Requests\_Parts.Goods\_ID order by [Request\_Date] desc |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Characteristics] (  [ID\_Characteristics] [int] not null identity(1,1),  [Goods\_ID] [int] not null,  [Characteristic\_Name\_ID] [int] not null,  [Characteristics \_Value] [varchar] (50) not null  constraint [PK\_Characteristics] primary key clustered  ([ID\_Characteristics] ASC) on [PRIMARY],  constraint [FK\_Characteristics\_Goods] foreign key ([Goods\_ID])  references [dbo].[Goods] ([ID\_Goods]),  constraint [FK\_Characteristics\_Characteristic\_Name] foreign key ([Characteristic\_Name\_ID])  references [dbo].[Characteristics\_Name] ([ID\_Characteristic\_Name]) ) go | - |
| Запросы на добавление данных |
| insert into [dbo].[Characteristics] ([Goods\_ID], [Characteristic\_Name\_ID], [Characteristics\_Value]) values (1, 4, '2520'); go |
| Запросы на выборку данных |
| Задача: Вывести название и артикул товара, название характеристики и её значение , отсортировав по названию и артикулу товара |
| select [Goods\_Name] + ' ' + [Goods\_Article] as 'Товар', [Characteristic\_Name] as 'Название характеристики',  [Characteristics\_Value] as 'Значение характеристики' from [dbo].[Characteristics] join Characteristics\_Name CN on CN.ID\_Characteristic\_Name = Characteristics.Characteristic\_Name\_ID join Goods G on G.ID\_Goods = Characteristics.Goods\_ID order by [Товар] asc |
| Результат запроса на выборку данных | |
|  | |

1. Изменение и удаление данных в таблицах

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Country] (  [ID\_Country] [int] not null identity(1,1),  [Country\_Name] [varchar] (50) not null  constraint [PK\_Country] primary key clustered  ([ID\_Country] ASC) on [PRIMARY],  constraint [UQ\_Country\_Name] unique ([Country\_Name]) ) go | Изменение данных |
| update [dbo].[Country] set  [Country\_Name] = @Country\_Name  where [ID\_Country] = @ID\_Country |
|  |
| Удаление данных |
| delete from [dbo].[Country] where ID\_Country = 1 go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Manufacturer] (  [ID\_Manufacturer] [int] not null identity(1,1),  [Manufacturer\_Name] [varchar] (50) not null,  [Manufacturer\_Country\_ID] [int] not null  constraint [PK\_Manufacturer] primary key clustered  ([ID\_Manufacturer] ASC) on [PRIMARY],  constraint [UQ\_Manufacturer\_Name] unique ([Manufacturer\_Name]),  constraint [FK\_Country\_Manufacturer] foreign key ([Manufacturer\_Country\_ID])  references [dbo].[Country] ([ID\_Country]) ) go | Изменение данных |
| update [dbo].[Manufacturer] set  [Manufacturer\_Name] = 'ASUS',  [Manufacturer\_Country\_ID] = 2  where [ID\_Manufacturer] = 2 go |
|  |
| Удаление данных |
| delete from [dbo].[Manufacturer] where ID\_Manufacturer = 5 go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Buyer] (  [ID\_Buyer] [int] not null identity(1,1),  [Buyer\_Login] [varchar] (32) not null,  [Buyer\_Password] [varchar] (32) not null,  [Buyer\_Phone\_Number] [varchar] (17) not null,  [Buyer\_Address] [varchar] (max) not null,  constraint [PK\_Buyer] primary key clustered  ([ID\_Buyer] ASC) on [PRIMARY],  constraint [UQ\_Buyer\_Login] unique ([Buyer\_Login]),  constraint [UQ\_Buyer\_Phone\_Number] unique ([Buyer\_Phone\_Number]),  constraint [CH\_Buyer\_Login] check (*len*([Buyer\_Login])>=8),  constraint [CH\_Buyer\_Password\_Upper] check ([Buyer\_Password] like ('%[A-Z]%')),  constraint [CH\_Buyer\_Password\_Lower] check ([Buyer\_Password] like ('%[a-z]%')),  constraint [CH\_Buyer\_Password\_Symbols] check ([Buyer\_Password] like ('%[!@#$%^&\*()]%')),  constraint [CH\_Buyer\_Phone\_Number] check ([Buyer\_Phone\_Number] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]')  ) go | Изменение данных |
| update [dbo].[Buyer] set  [Buyer\_Login] = 'Userbebra1\_1',  [Buyer\_Password] = 'R@0t\_1',  [Buyer\_Phone\_Number] = '+7(999)999-99-99',  [Buyer\_Address] = '123456, Владимирская область, город Подольск, пл. Славы, 23'  where [ID\_Buyer] = 2 go |
|  |
| Удаление данных |
| delete from [dbo].[Buyer] where ID\_Buyer = 2 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,  [Card\_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\_Card\_Holder] check ([Card\_Holder] = *UPPER*([Card\_Holder])),  constraint [CH\_Card\_Validity] check ([Card\_Validity] like ('[0-1][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\_Number] = '3333 3333 3333 3333',  [Card\_Holder] = 'THIRD BUYER',  [Card\_Validity] = '03/03',  [Buyer\_ID] = 4  where [ID\_Buyer\_Card] = 3 go |
|  |
| Удаление данных |
| delete from [dbo].[Buyer\_Card] where ID\_Buyer\_Card = 1 go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Supplier] (  [ID\_Supplier] [int] not null identity(1,1),  [Supplier\_Full\_Name] [varchar] (max) not null,  [Supplier\_Short\_Name] [varchar] (50) not null,  [Supplier\_Legal\_Address] [varchar] (max) not null,  [Supplier\_Physical\_Address] [varchar] (max) not null,  [Supplier\_E\_Mail] [varchar] (max) not null,  [Supplier\_Phone\_Number] [varchar] (17) not null  constraint [PK\_Supplier] primary key clustered  ([ID\_Supplier] ASC) on [PRIMARY],  constraint [UQ\_Supplier\_Short\_Name] unique ([Supplier\_Short\_Name]),  constraint [UQ\_Supplier\_Phone\_Number] unique ([Supplier\_Phone\_Number]),  constraint [CH\_Supplier\_E\_Mail] check ([Supplier\_E\_Mail] like ('%@%.%')),  constraint [CH\_Supplier\_Phone\_Number] check ([Supplier\_Phone\_Number] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]') ) go | Изменение данных |
| update [dbo].[Supplier] set  [Supplier\_Full\_Name] = 'Shiza\_Bebra\_Purinda',  [Supplier\_Short\_Name] = 'SBP',  [Supplier\_Legal\_Address] = 'Пакистан, 7755',  [Supplier\_Physical\_Address] = 'Галакика Андромеда',  [Supplier\_E\_Mail] = 'wtfis@this.email',  [Supplier\_Phone\_Number] = '+7(516)171-81-92'  where [ID\_Supplier] = 2 go |
|  |
| Удаление данных |
| delete from [dbo].[Supplier] where ID\_Supplier = 1 go |
|  |

1. Фильтрация данных

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Representative] (  [ID\_Representative] [int] not null identity(1,1),  [Supplier\_ID] [int] not null,  [Repres\_First\_Name] [varchar] (30) not null,  [Repres\_Second\_Name] [varchar] (30) not null,  [Repres\_Middle\_Name] [varchar] (30) null default('-')  constraint [PK\_Representative] primary key clustered  ([ID\_Representative] ASC) on [PRIMARY],  constraint [FK\_Supplier\_Representative] foreign key ([Supplier\_ID])  references [dbo].[Supplier] ([ID\_Supplier]) ) go | Задача: Вывести фио представителей поставщика с кратким названием ‘CUM’ |
| select ([Repres\_Second\_Name] + ' ' + [Repres\_First\_Name] + ' ' + [Repres\_Middle\_Name])  as 'ФИО представителя' from [dbo].[Representative] join Supplier S on S.ID\_Supplier = Representative.Supplier\_ID where [Supplier\_Short\_Name] = 'CUM' |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Contract] (  [ID\_Contract] [int] not null identity(1,1),  [Representative\_ID] [int] not null,  [Contract\_Expiry\_Date] [date] not null,  [Contract\_Conclusion\_Date] [date] null default(*getdate*())  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [CH\_Contract\_Expiry\_Date] check ([Contract\_Expiry\_Date] >= *CAST*(*getdate*() as date)),  constraint [CH\_Contract\_Conclusion\_Date] check ([Contract\_Conclusion\_Date] <= *CAST*(*getdate*() as date)),  constraint [FK\_Representative\_Contract] foreign key ([Representative\_ID])  references [dbo].[Representative] ([ID\_Representative]) ) go | Задача: Вывести краткие названия поставщиков с которыми на данный моент есть действующие контракты |
| select DISTINCT([Supplier\_Short\_Name]) from [dbo].[Contract]  join Representative on Representative.ID\_Representative = Contract.Representative\_ID  join Supplier S on S.ID\_Supplier = Representative.Supplier\_ID where Contract\_Expiry\_Date > *getdate*() |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Characteristics\_Name] (  [ID\_Characteristic\_Name] [int] not null identity(1,1),  [Characteristic\_Name] [varchar] (30) not null  constraint [PK\_Characteristic\_Name] primary key clustered  ([ID\_Characteristic\_Name] ASC) on [PRIMARY],  constraint [UQ\_Characteristic\_Name] unique ([Characteristic\_Name]) ) go | Задача: Вывести все названия характеристик начинающихся с «Частота» |
| select [Characteristic\_Name] as 'Название характеристия' from [dbo].[Characteristics\_Name] where [Characteristic\_Name] like 'Частота%' order by [Characteristic\_Name] asc |
|  |