**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].[Employee]  (  [ID\_Employee] [int] not null identity(1,1),  [Name\_Employee] [varchar](50) not null,  [Surname\_Employee] [varchar](50) not null,  [Patronymic\_Employee] [varchar](50) null default 'Нет данных',  [Login\_Employee] [varchar](255) not null,  [Password\_Employee] [varchar](36) not null,  constraint [PK\_Employee] primary key clustered  ([ID\_Employee] ASC) on [PRIMARY],  constraint [UQ\_Login\_Employee] unique ([Login\_Employee]),  constraint [CH\_Login\_Employee] check (len([Login\_Employee]) >= 6),  constraint [CH\_Password\_Employee] check (len([Password\_Employee]) >= 6 and [Password\_Employee] like '%[a-zA-Z]%' and [Password\_Employee] like '%[0-9]%' and [Password\_Employee] like '%[!@#$%^&\*()]%')  )  go | - |
| Запросы на добавление данных |
| INSERT INTO [dbo].[Employee] ([Name\_Employee], [Surname\_Employee], [Patronymic\_Employee], [Login\_Employee], [Password\_Employee])  VALUES  ('Иван', 'Иванов', 'Иванович', 'em\_IvanovII', 'Pa$$w0rd'),  ('Пётр', 'Петров', 'Петрович', 'em\_PetrovPP', 'Pa$$w0rd'),  ('Алексей', 'Алексеев', 'Алексеевич', 'em\_Alekseev', 'Pa$$w0rd');  GO |
| Запросы на выборку данных |
| Задача: Вывести, отсортировав фамилии по возрастанию. |
| SELECT  [Name\_Employee] + ' ' + [Surname\_Employee] + ' ' + [Patronymic\_Employee] AS FullName,  [Login\_Employee],  [Password\_Employee]  FROM [dbo].[Employee]  ORDER BY [Surname\_Employee] ASC;  GO |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Client]  (  [ID\_Client] [int] not null identity(1,1),  [Surname\_Client] [varchar](50) not null,  [Name\_Client] [varchar](50) not null,  [Patronymic\_Client] [varchar](50) null default 'Нет данных',  [Insurance\_Number\_Client] [varchar](19) not null,  [Passport\_Series\_Client] [varchar](5) not null,  [Passport\_Number\_Client] [varchar](6) not null,  [Birth\_Client] [datetime] not null,  [Gender\_Client] [varchar](3) not null,  [Who\_Issued\_Passport\_Client] [varchar](max) not null,  [When\_Issued\_Passport\_Client] [date] not null,  [Division\_Code\_Passport\_Client] [varchar](7) not null,  [Phone\_Number\_Client] [varchar](16) not null,  [Email\_Client] [varchar](255) not null,  [Login\_Client] [varchar](255) not null,  [Password\_Client] [varchar](36) not null,  constraint [PK\_Client] primary key clustered  ([ID\_Client] ASC) on [PRIMARY],  constraint [UQ\_Insurance\_Number\_Client] unique ([Insurance\_Number\_Client]),  constraint [UQ\_Phone\_Number\_Client] unique ([Phone\_Number\_Client]),  constraint [UQ\_Email\_Client] unique ([Email\_Client]),  constraint [UQ\_Login\_Client] unique ([Login\_Client]),  constraint [CH\_Insurance\_Number\_Client] check ([Insurance\_Number\_Client] 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\_Passport\_Series\_Client] check ([Passport\_Series\_Client] like '[0-9][0-9] [0-9][0-9]'),  constraint [CH\_Passport\_Number\_Client] check ([Passport\_Number\_Client] like '[0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Birth\_Client] check ([Birth\_Client] <= dateadd(year, -17, getdate())),  constraint [CH\_Gender\_Client] check ([Gender\_Client] in ('Муж', 'Жен')),  constraint [CH\_When\_Issued\_Passport\_Client] check ([When\_Issued\_Passport\_Client] >= dateadd(year, -14, getdate())),  constraint [CH\_Division\_Code\_Passport\_Client] check ([Division\_Code\_Passport\_Client] like '[0-9][0-9][0-9]-[0-9][0-9][0-9]'),  constraint [CH\_Phone\_Number\_Client] check ([Phone\_Number\_Client] like '+7([0-9][0-9][0-9])[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),  constraint [CH\_Email\_Client] check ([Email\_Client] like '%\_@\_\_%.\_\_%'),  constraint [CH\_Login\_Client] check (len([Login\_Client]) >= 6 and [Login\_Client] like '%[a-zA-Z]%'),  constraint [CH\_Password\_Client] check (len([Password\_Client]) >= 6 and [Password\_Client] like '%[a-zA-Z]%' and [Password\_Client] like '%[0-9]%' and [Password\_Client] like '%[!@#$%^&\*()]%')  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Client] (  [Surname\_Client],  [Name\_Client],  [Patronymic\_Client],  [Insurance\_Number\_Client],  [Passport\_Series\_Client],  [Passport\_Number\_Client],  [Birth\_Client],  [Gender\_Client],  [Who\_Issued\_Passport\_Client],  [When\_Issued\_Passport\_Client],  [Division\_Code\_Passport\_Client],  [Phone\_Number\_Client],  [Email\_Client],  [Login\_Client],  [Password\_Client]  )  values  ('Андреев', 'Алексей', 'Дмитриевич', '0001 0001 1008 1999', '45 92', '642742', CONVERT(date, '1999-08-10', 120), 'Муж', 'Отделение УФМС России по городу Москве по ЮАО', CONVERT(date, '2019-09-11', 120), '770-262', '+7(945)762-57-92', 'Andreev\_a\_a@gmail.com', 'AndreevAD', 'Pa$$w0rd'),  ('Владимиров', 'Евгений', 'Олегович', '0000 0001 2402 1998', '44 65', '212964', CONVERT(date,'1998-02-24', 120), 'Муж', 'Отделение УФМС России по городу Москве по СВАО', CONVERT(date, '2018-03-25', 120), '770-912', '+7(955)874-63-02', 'Vladimirov\_e\_o@ya.ru', 'VladimirovEO', 'Pa$$w0rd'),  ('Никитин', 'Кирилл', 'Семёнович', '0001 0000 0711 2001', '45 12', '633276', CONVERT(date, '2001-11-07', 120), 'Муж', 'Отделение УФМС России по городу Москве по САО', CONVERT(date, '2021-12-08', 120), '770-124', '+7(947)587-36-42', 'Nikitin\_k\_s@gmail.com', 'NikitinKS', 'Pa$$w0rd'),  ('Павлов', 'Леонид', 'Андреевич', '0001 0111 2010 2000', '48 31', '874337', CONVERT(date, '2000-10-20', 120), 'Муж', 'Отделение УФМС России по городу Москве по ВАО', CONVERT(date, '2020-11-21', 120), '770-982', '+7(995)475-03-54', 'Pavlov\_l\_a@ya.ru', 'PavlovLA', 'Pa$$w0rd'),  ('Иванов', 'Павел', 'Александрович', '0000 0001 2905 2002', '42 66', '167832', CONVERT(date, '2002-05-29', 120), 'Муж', 'Отделение УФМС России по городу Москве по ЮАО', CONVERT(date, '2022-06-30', 120), '770-262', '+7(982)716-52-12', 'Ivanov\_p\_a@ya.ru', 'IvanovPA', 'Pa$$w0rd'),  ('Романов', 'Дмитрий', 'Антонович', '0100 1000 0109 2001', '43 56', '843377', CONVERT(date, '2001-08-01', 120), 'Муж', 'Отделение УФМС России по городу Москве по ЦАО', CONVERT(date, '2021-09-02', 120), '770-832', '+7(935)768-23-79', 'Ronmanov\_d\_a@ya.ru', 'RomanovDA', 'Pa$$w0rd'),  ('Егоров', 'Игорь', 'Олегович', '0000 0000 1807 2000', '45 67', '113456', CONVERT(date, '2000-07-18', 120), 'Муж', 'Отделение УФМС России по городу Москве по ЦАО', CONVERT(date, '2020-08-19', 120), '770-832', '+7(993)557-66-57', 'Egorov\_i\_o@gmail.com', 'EgorovIO', 'Pa$$w0rd');  go |
| Запросы на выборку данных |
| Задача: Вывести данные клиентов, отсортировав по ФИО. |
| select  [Surname\_Client] + ' ' + [Name\_Client] + ' ' + [Patronymic\_Client] as [ФИО],  [Insurance\_Number\_Client] as [Номер страховки],  [Passport\_Series\_Client] + ' ' + [Passport\_Number\_Client] as [Паспорт],  [Birth\_Client] as [Дата рождения],  [Gender\_Client] as [Пол],  [Who\_Issued\_Passport\_Client] as [Кем выдан паспорт],  [When\_Issued\_Passport\_Client] as [Когда выдан паспорт],  [Division\_Code\_Passport\_Client] as [Код подразделения],  [Phone\_Number\_Client] as [Телефон],  [Email\_Client] as [Email],  [Login\_Client] as [Логин],  [Password\_Client] as [Пароль]  from [dbo].[Client]  order by [Surname\_Client], [Name\_Client], [Patronymic\_Client];  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Map]  (  [ID\_Map] [int] not null identity(1,1),  [Name\_Map] [varchar](20) not null,  [Square\_Map] [decimal](5,2) not null,  [Length\_Map] [decimal](5,2) not null,  [Width\_Map] [decimal](5,2) not null,  [Plan\_Map] [varchar](max) not null,  [Buildings\_Map] [varchar](max) null default 'Отсутствует',  [Description\_Map] [varchar](max) not null,  [Cost\_Map] [decimal](10,3) not null,  constraint [PK\_Map] primary key clustered  ([ID\_Map] ASC) on [PRIMARY],  constraint [UQ\_Name\_Map] unique ([Name\_Map]),  constraint [CH\_Square\_Map] check ([Square\_Map] > 0),  constraint [CH\_Length\_Map] check ([Length\_Map] > 0),  constraint [CH\_Width\_Map] check ([Width\_Map] > 0),  constraint [CH\_Cost\_Map] check ([Cost\_Map] >= 0)  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Map] (  [Name\_Map],  [Square\_Map],  [Length\_Map],  [Width\_Map],  [Plan\_Map],  [Buildings\_Map],  [Description\_Map],  [Cost\_Map]  )  values  ('Карта-А1', 247.5, 15.0, 16.5, 'C:\Users\Планы карт\Карта-А1.png', 'Четыре отдельных одноэтажных сооружений', 'Открытая местность, по краям лесополоса', 2500.00),  ('Карта-А2', 130.0, 10.0, 13.0, 'C:\Users\Планы карт\Карта-А2.png', 'Отсутствуют', 'Холмистая местность, с небольшим водоёмом', 2000.00),  ('Карта-Б1', 203.7, 21.0, 9.7, 'C:\Users\Планы карт\Карта-Б1.png', 'Одно большое здание на три этажа', 'Общее возвышение рельефа с запада на восток на 4 метра', 2750.00),  ('Карта-Б2', 104.0, 13.0, 8.0, 'C:\Users\Планы карт\Карта-Б2.png', 'Два здания высотой в два этажа', 'Равнина, преобладает редкий лес', 2500.00);  go |
| Запросы на выборку данных |
| Задача: Вывести данные, отсортировав имена по алфавиту. |
| select  [Name\_Map],  [Square\_Map],  [Length\_Map],  [Width\_Map],  [Plan\_Map],  [Buildings\_Map],  [Description\_Map],  [Cost\_Map]  from [dbo].[Map]  order by [Name\_Map];  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Game\_Type]  (  [ID\_Game\_Type] [int] not null identity(1,1),  [Name\_Game\_Type] [varchar](20) not null,  [Cost\_Game\_Type] [decimal](10,3) not null,  [Description\_Game\_Type] [varchar](max) not null,  constraint [PK\_Game\_Type] primary key clustered  ([ID\_Game\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Game\_Type] unique ([Name\_Game\_Type]),  constraint [CH\_Cost\_Game\_Type] check ([Cost\_Game\_Type] >= 0)  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Game\_Type] (  [Name\_Game\_Type],  [Cost\_Game\_Type],  [Description\_Game\_Type]  )  values  ('Все против всех', 1000.00, 'Все игроки набивают очки за попадание друг в друга'),  ('Командный матч', 1100.00, 'Игроки делятся на команды и набивают очки за попадание в команду противника'),  ('На выбывание', 900.00, 'Игра длится до тех пор, пока, не останется последний «Чистый» игрок'),  ('Штурм', 1200.00, 'Есть атакующие и защищающие. Атакующие штурмуют объект, который защищает другая команда');  go |
| Запросы на выборку данных |
| **Задача:** Вывести название типа игры, стоимость и описание, отсортировав название типа игры по возрастанию. |
| select [Name\_Game\_Type] as "Название типа игры", [Cost\_Game\_Type] as "Стоимость игры", [Description\_Game\_Type] as "Описание игры"  from [dbo].[Game\_Type]  order by [Name\_Game\_Type] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Inventory]  (  [ID\_Inventory] [int] not null identity(1,1),  [Number\_Inventory] [varchar](16) not null,  [Count\_Inventory] [int] not null,  [Cost\_Inventory] [decimal](10,3) not null,  constraint [PK\_Inventory] primary key clustered  ([ID\_Inventory] ASC) on [PRIMARY],  constraint [UQ\_Number\_Inventory] unique ([Number\_Inventory]),  constraint [CH\_Number\_Inventory] check ([Number\_Inventory] like 'ИнВ-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Count\_Inventory] check ([Count\_Inventory] >= 0),  constraint [CH\_Cost\_Inventory] check ([Cost\_Inventory] >= 0)  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Inventory] (  [Number\_Inventory],  [Count\_Inventory],  [Cost\_Inventory]  )  values  ('ИнВ-00000001', 20, 1500.00),  ('ИнВ-00000002', 15, 1650.00),  ('ИнВ-00000003', 21, 2000.00),  ('ИнВ-00000004', 125, 150.00),  ('ИнВ-00000005', 130, 200.00),  ('ИнВ-00000006', 30, 350.00),  ('ИнВ-00000007', 30, 650.00),  ('ИнВ-00000008', 60, 200.00),  ('ИнВ-00000009', 70, 250.00);  go |
| Запросы на выборку данных |
| **Задача:** Вывести номер инвентаря, количество и стоимость, отсортировав данные по возрастанию по номеру инвентаря. |
| select [Number\_Inventory] as "Номер инвентаря", [Count\_Inventory] as "Количество", [Cost\_Inventory] as "Стоимость"  from [dbo].[Inventory]  order by [Number\_Inventory] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Characteristic]  (  [ID\_Characteristic] [int] not null identity(1,1),  [Value\_Characteristic] [varchar](50) not null,  constraint [PK\_Characteristic] primary key clustered  ([ID\_Characteristic] ASC) on [PRIMARY],  constraint [UQ\_Value\_Characteristic] unique ([Value\_Characteristic])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Characteristic] (  [Value\_Characteristic]  )  values  ('Стальной ствол'),  ('Одиночный режим стрельбы'),  ('Стрельба очередями'),  ('Карбоновый ствол'),  ('Легче на 250 гр'),  ('Комбинированный режим стрельбы'),  ('Тонкая оболочка'),  ('Одно цвета'),  ('Разноцветные'),  ('Легкий материал'),  ('Защищает всё лицо'),  ('Толстый слой металла'),  ('Средней толщины слой металла');  go |
| Запросы на выборку данных |
| **Задача:** Вывести все характеристики, отсортировав их по возрастанию. |
| select [Value\_Characteristic] as "Характеристика"  from [dbo].[Characteristic]  order by [Value\_Characteristic] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Contract]  (  [Employee\_ID] [int] not null,  [Game\_Type\_ID] [int] not null,  [Map\_ID] [int] not null,  [ID\_Contract] [int] not null identity(1,1),  [Number\_Contract] [varchar](10) not null,  [Cost\_Contract] [decimal](10,3) not null,  [DateTime\_Forming\_Contract] [datetime] not null,  [Game\_Duration\_Contract] [decimal](5,2) not null,  [Game\_Start\_Time\_Contract] [time] not null,  [Game\_End\_Time\_Contract] [time] not null,  [Final\_Cost\_Contract] [decimal](10,3) not null,  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [UQ\_Number\_Contract] unique ([Number\_Contract]),  constraint [CH\_Number\_Contract] check ([Number\_Contract] like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Cost\_Contract] check ([Cost\_Contract] >= 0),  constraint [CH\_Game\_Duration\_Contract] check ([Game\_Duration\_Contract] >= 0),  constraint [CH\_Game\_Start\_Time\_Contract] check ([Game\_Start\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Game\_End\_Time\_Contract] check ([Game\_End\_Time\_Contract] > [Game\_Start\_Time\_Contract] and [Game\_End\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Final\_Cost\_Contract] check ([Final\_Cost\_Contract] > 0),  constraint [FK\_Employee\_Contract] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [FK\_Game\_Type\_Contract] foreign key ([Game\_Type\_ID])  references [dbo].[Game\_Type] ([ID\_Game\_Type]),  constraint [FK\_Map\_Contract] foreign key ([Map\_ID])  references [dbo].[Map] ([ID\_Map])  )  go | insert into [dbo].[Contract] (  [Employee\_ID],  [Game\_Type\_ID],  [Map\_ID],  [Number\_Contract],  [Cost\_Contract],  [DateTime\_Forming\_Contract],  [Game\_Duration\_Contract],  [Game\_Start\_Time\_Contract],  [Game\_End\_Time\_Contract],  [Final\_Cost\_Contract]  )  values  (1, 2, 4, 'GM-0000001', 2000.00, CONVERT(date, '2023-09-10 10:40', 120), 2.0, '11:00', '13:00', 9300.00),  (1, 4, 4, 'GM-0000002', 1500.00, CONVERT(date, '2023-10-14 11:30', 120), 1.5, '12:00', '13:30', 7000.00),  (2, 2, 1, 'GM-0000003', 2000.00, CONVERT(date, '2023-10-20 09:43', 120), 2.0, '12:15', '14:15', 8500.00);  go |
| Запросы на добавление данных |
|  |
| Запросы на выборку данных |
| **Задача:** Вывести номер контракта, стоимость, дату и время формирования, длительность игры, время начала и окончания игры, а также итоговую стоимость контракта, отсортировав данные по возрастанию по номеру контракта. |
| select  [Number\_Contract] as "Номер контракта",  [Cost\_Contract] as "Стоимость контракта",  [DateTime\_Forming\_Contract] as "Дата и время формирования",  [Game\_Duration\_Contract] as "Длительность игры (часы)",  [Game\_Start\_Time\_Contract] as "Время начала игры",  [Game\_End\_Time\_Contract] as "Время окончания игры",  [Final\_Cost\_Contract] as "Итоговая стоимость контракта"  from [dbo].[Contract]  order by [Number\_Contract] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Player\_Type]  (  [ID\_Player\_Type] [int] not null identity(1,1),  [Name\_Player\_Type] [varchar](20) not null,  constraint [PK\_Player\_Type] primary key clustered  ([ID\_Player\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Player\_Type] unique ([Name\_Player\_Type]),  constraint [CH\_Name\_Player\_Type] check ([Name\_Player\_Type] in ('Ответственный', 'Участник'))  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Player\_Type] (  [Name\_Player\_Type]  )  values  ('Участник'),  ('Ответственный');  go |
| Запросы на выборку данных |
| **Задача:** Вывести название типа игрока, отсортировав по возрастанию. |
| select  [Name\_Player\_Type] as "Тип игрока"  from [dbo].[Player\_Type]  order by [Name\_Player\_Type] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Client\_Contract]  (  [ID\_Client\_Contract] [int] not null identity(1,1),  [Contract\_ID] [int] not null,  [Client\_ID] [int] not null,  [Player\_Type\_ID] [int] not null,  constraint [PK\_Client\_Contract] primary key clustered  ([ID\_Client\_Contract] ASC) on [PRIMARY],  constraint [FK\_Contract\_Client\_Contract] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract]),  constraint [FK\_Client\_Client\_Contract] foreign key ([Client\_ID])  references [dbo].[Client] ([ID\_Client]),  constraint [FK\_Player\_Type\_Client\_Contract] foreign key ([Player\_Type\_ID])  references [dbo].[Player\_Type] ([ID\_Player\_Type])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Client\_Contract] (  [Contract\_ID],  [Client\_ID],  [Player\_Type\_ID]  )  values  (1, 1, 1),  (1, 2, 1),  (1, 5, 1),  (1, 3, 2),  (2, 4, 1),  (2, 5, 1),  (2, 1, 1),  (3, 7, 2),  (3, 2, 1),  (3, 7, 1),  (3, 3, 2);  go |
| Запросы на выборку данных |
| **Задача:** Вывести номер договора, идентификатор клиента и тип игрока, отсортировав по возрастанию по номеру договора. |
| select  [Contract\_ID] as "Номер договора",  [Client\_ID] as "Клиент",  [Player\_Type\_ID] as "Тип игрока"  from [dbo].[Client\_Contract]  order by [Contract\_ID] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Inventory\_Characteristic]  (  [ID\_Inventory\_Characteristic] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Characteristic\_ID] [int] not null,  constraint [PK\_Inventory\_Characteristic] primary key clustered  ([ID\_Inventory\_Characteristic] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Inventory\_Characteristic] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Characteristic\_Inventory\_Characteristic] foreign key ([Characteristic\_ID])  references [dbo].[Characteristic] ([ID\_Characteristic])  )  go |  |
| Запросы на добавление данных |
| insert into [dbo].[Inventory\_Characteristic] (  [Inventory\_ID],  [Characteristic\_ID]  )  values  (1, 1),  (1, 2),  (2, 1),  (2, 3),  (3, 4),  (3, 5),  (3, 6),  (4, 7),  (4, 8),  (5, 7),  (5, 9),  (6, 10),  (6, 11),  (7, 12),  (8, 13),  (9, 13);  go |
| Запросы на выборку данных |
| **Задача:** Вывести идентификатор инвентаря и идентификатор характеристики, отсортировав по возрастанию по идентификатору инвентаря. |
| select  [Inventory\_ID] as "ID Инвентаря",  [Characteristic\_ID] as "ID Характеристики"  from [dbo].[Inventory\_Characteristic]  order by [Inventory\_ID] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Inventory\_Contract]  (  [ID\_Inventory\_Contract] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Contract\_ID] [int] not null,  constraint [PK\_Inventory\_Contract] primary key clustered  ([ID\_Inventory\_Contract] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Inventory\_Contract] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Contract\_Inventory\_Contract] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Inventory\_Contract] (  [Inventory\_ID],  [Contract\_ID]  )  values  (1, 1),  (1, 1),  (5, 1),  (6, 2),  (6, 2),  (2, 3),  (9, 3);  go |
| Запросы на выборку данных |
| **Задача:** Вывести идентификаторы инвентаря и договоров, отсортировав по возрастанию по идентификатору инвентаря. |
| select  [Inventory\_ID] as "ID Инвентаря",  [Contract\_ID] as "ID Договора"  from [dbo].[Inventory\_Contract]  order by [Inventory\_ID] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
|  | - |
| Запросы на добавление данных |
|  |
| Запросы на выборку данных |
|  |
|  |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Additional\_Client\_Contract]  (  [ID\_Additional\_Client\_Contract] [int] not null identity(1,1),  [Number\_Additional\_Contract] [varchar](12) not null,  [Client\_Contract\_ID] [int] not null,  constraint [PK\_Additional\_Client\_Contract] primary key clustered  ([ID\_Additional\_Client\_Contract] ASC) on [PRIMARY],  constraint [UQ\_Number\_Additional\_Contract] unique ([Number\_Additional\_Contract]),  constraint [CH\_Number\_Additional\_Contract] check ([Number\_Additional\_Contract] like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]/[0-9]'),  constraint [FK\_Client\_Contract\_Additional\_Client\_Contract] foreign key ([Client\_Contract\_ID])  references [dbo].[Client\_Contract] ([ID\_Client\_Contract])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Additional\_Client\_Contract] (  [Client\_Contract\_ID],  [Number\_Additional\_Contract]  )  values  (1, 'GM-0000001/1'),  (2, 'GM-0000001/2'),  (3, 'GM-0000001/3'),  (5, 'GM-0000002/1'),  (6, 'GM-0000002/2'),  (7, 'GM-0000002/3'),  (9, 'GM-0000003/1'),  (10, 'GM-0000003/2');  go |
| Запросы на выборку данных |
| **Задача:** Вывести номера дополнительных договоров и идентификаторы клиентских договоров, отсортировав по возрастанию по номеру дополнительного договора. |
| select  [Client\_Contract\_ID] as "ID Клиентского договора",  [Number\_Additional\_Contract] as "Номер дополнительного договора"  from [dbo].[Additional\_Client\_Contract]  order by [Number\_Additional\_Contract] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Armor\_Type]  (  [ID\_Armor\_Type] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Name\_Armor\_Type] [varchar](15) not null,  constraint [PK\_Armor\_Type] primary key clustered  ([ID\_Armor\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Armor\_Type] unique ([Name\_Armor\_Type]),  constraint [FK\_Inventory\_Armor\_Type] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Armor\_Type] (  [Inventory\_ID],  [Name\_Armor\_Type]  )  values  (6, 'Шлем'),  (7, 'Бронежилет'),  (8, 'Налокотники'),  (9, 'Наколенники');  go |
| Запросы на выборку данных |
| **Задача:** Вывести названия видов брони и идентификаторы инвентаря, отсортировав по возрастанию названий вида брони. |
| select  [Inventory\_ID] as "ID Инвентаря",  [Name\_Armor\_Type] as "Название вида брони"  from [dbo].[Armor\_Type]  order by [Name\_Armor\_Type] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Additional\_Inventory]  (  [ID\_Additional\_Inventory] [int] not null identity(1,1),  [Name\_Additional\_Inventory] [varchar](30) not null,  [Count\_Additional\_Inventory] [int] not null,  [Cost\_Additional\_Inventory] [decimal](10,3) not null,  [Contract\_ID] [int] not null,  constraint [PK\_Additional\_Inventory] primary key clustered  ([ID\_Additional\_Inventory] ASC) on [PRIMARY],  constraint [CH\_Count\_Additional\_Inventory] check ([Count\_Additional\_Inventory] >= 0),  constraint [CH\_Cost\_Additional\_Inventory] check ([Cost\_Additional\_Inventory] >= 0),  constraint [FK\_Contract\_Additional\_Inventory] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Additional\_Inventory] (  [Contract\_ID],  [Name\_Additional\_Inventory],  [Count\_Additional\_Inventory],  [Cost\_Additional\_Inventory]  )  values  (1, 'Блок защиты', 2, 500),  (2, 'Блок защиты', 2, 500),  (2, 'Металлические бочки', 6, 600),  (3, 'Металлические бочки', 10, 1000);  go |
| Запросы на выборку данных |
| **Задача:** Вывести названия дополнительного инвентаря, количество и стоимость, отсортировав по возрастанию названий. |
| select  [Contract\_ID] as "ID Договора",  [Name\_Additional\_Inventory] as "Название дополнительного инвентаря",  [Count\_Additional\_Inventory] as "Количество",  [Cost\_Additional\_Inventory] as "Стоимость"  from [dbo].[Additional\_Inventory]  order by [Name\_Additional\_Inventory] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Mark]  (  [ID\_Mark] [int] not null identity(1,1),  [Name\_Mark] [varchar](15) not null,  constraint [PK\_Mark] primary key clustered  ([ID\_Mark] ASC) on [PRIMARY],  constraint [UQ\_Name\_Mark] unique ([Name\_Mark])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Mark] (  [Name\_Mark]  )  values  ('MarkerShot'),  ('Body Arm');  go |
| Запросы на выборку данных |
| **Задача:** Вывести названия марок, отсортировав по возрастанию. |
| select  [Name\_Mark] as "Название марки"  from [dbo].[Mark]  order by [Name\_Mark] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Model]  (  [ID\_Model] [int] not null identity(1,1),  [Name\_Model] [varchar](15) not null,  [Mark\_ID] [int] not null,  constraint [PK\_Model] primary key clustered  ([ID\_Model] ASC) on [PRIMARY],  constraint [UQ\_Name\_Model] unique ([Name\_Model]),  constraint [FK\_Mark\_Model] foreign key ([Mark\_ID])  references [dbo].[Mark] ([ID\_Mark])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Model] (  [Mark\_ID],  [Name\_Model]  )  values  (1, 'Mrk I'),  (1, 'Mrk I-I'),  (1, 'Mrk II'),  (1, 'PlasticBall 100'),  (1, 'PlasticBall 90'),  (2, 'Hlmt-100'),  (2, 'Shld-130'),  (2, 'HdsDf-10'),  (2, 'LgsDf-10');  go |
| Запросы на выборку данных |
| **Задача:** Вывести идентификаторы марок и названия моделей, отсортировав по возрастанию названий моделей. |
| select  [Mark\_ID] as "ID Марки",  [Name\_Model] as "Название модели"  from [dbo].[Model]  order by [Name\_Model] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Mark\_Model\_Inventory]  (  [ID\_Mark\_Model\_Inventory] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Model\_ID] [int] not null,  constraint [PK\_Mark\_Model\_Inventory] primary key clustered  ([ID\_Mark\_Model\_Inventory] ASC) on [PRIMARY],  constraint [FK\_Inventory\_Mark\_Model\_Inventory] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Model\_Mark\_Model\_Inventory] foreign key ([Model\_ID])  references [dbo].[Model] ([ID\_Model])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Mark\_Model\_Inventory] (  [Inventory\_ID],  [Model\_ID]  )  values  (1, 1),  (2, 2),  (3, 3),  (4, 4),  (5, 5),  (6, 6),  (7, 7),  (8, 8),  (9, 9);  go |
| Запросы на выборку данных |
| **Задача:** Вывести идентификаторы инвентаря и идентификаторы моделей, отсортировав по возрастанию идентификаторов моделей. |
| select  [Inventory\_ID] as "ID Инвентаря",  [Model\_ID] as "ID Модели"  from [dbo].[Mark\_Model\_Inventory]  order by [Model\_ID] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Suppliers]  (  [ID\_Supplier] [int] not null identity(1,1),  [Name\_Suppliers] [varchar](50) not null,  [OKPO\_Suppliers] [varchar](10) not null,  [Legal\_Address\_Suppliers] [varchar](max) not null,  [Physical\_Address\_Suppliers] [varchar](max) not null,  [Phone\_Number\_Suppliers] [varchar](16) not null,  constraint [PK\_Suppliers] primary key clustered  ([ID\_Supplier] ASC) on [PRIMARY],  constraint [UQ\_Name\_Suppliers] unique ([Name\_Suppliers]),  constraint [UQ\_OKPO\_Suppliers] unique ([OKPO\_Suppliers]),  constraint [UQ\_Phone\_Number\_Suppliers] unique ([Phone\_Number\_Suppliers]),  constraint [CH\_OKPO\_Suppliers] check ([OKPO\_Suppliers] like '[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Phone\_Number\_Suppliers] check ([Phone\_Number\_Suppliers] 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].[Suppliers] (  [Name\_Suppliers],  [OKPO\_Suppliers],  [Legal\_Address\_Suppliers],  [Physical\_Address\_Suppliers],  [Phone\_Number\_Suppliers]  )  values  ('ООО «Страйк-Прод»', '2434525672', 'г. Москва, ул. Новоясеневская, д. 10, к. 6', 'г. Москва, ул. Дмитровская, д. 19, стр. А', '+7(499)672-88-17'),  ('НПАО «Инвентарь Групп»', '9257257205', 'г. Москва, ул. Таганская, д. 54, стр. 1', 'г. Москва, ул. Коломенская, д. 11, стр. 8', '+7(499)106-76-95');  go |
| Запросы на выборку данных |
| **Задача:** Вывести информацию о поставщиках, отсортировав по названию поставщика. |
| select  [Name\_Suppliers] as "Название поставщика",  [OKPO\_Suppliers] as "ОКПО поставщика",  [Legal\_Address\_Suppliers] as "Юридический адрес",  [Physical\_Address\_Suppliers] as "Фактический адрес",  [Phone\_Number\_Suppliers] as "Телефон"  from [dbo].[Suppliers]  order by [Name\_Suppliers] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Order\_Status]  (  [ID\_Order\_Status] [int] not null identity(1,1),  [Status\_Order\_Status] [varchar](8) not null,  constraint [PK\_Order\_Status] primary key clustered  ([ID\_Order\_Status] ASC) on [PRIMARY],  constraint [UQ\_Status\_Order\_Status] unique ([Status\_Order\_Status]),  constraint [CH\_Status\_Order\_Status] check ([Status\_Order\_Status] in ('открыт', 'завершён'))  )  go |  |
| Запросы на добавление данных |
| insert into [dbo].[Order\_Status] (  [Status\_Order\_Status]  )  values  ('Открыт'),  ('Завершён');  go |
| Запросы на выборку данных |
| **Задача:** Вывести статусы заказа, отсортировав по статусу заказа в алфавитном порядке. |
| select  [Status\_Order\_Status] as "Статус заказа"  from [dbo].[Order\_Status]  order by [Status\_Order\_Status] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Supplier\_Contracts]  (  [ID\_Supplier\_Contracts] [int] not null identity(1,1),  [Supplier\_ID] [int] not null,  [Contract\_Number\_Supplier\_Contracts] [varchar](17) not null,  [Date\_Of\_Formation\_Supplier\_Contracts] [datetime] not null,  [Validity\_Period\_Supplier\_Contracts] [varchar](30) not null,  [Order\_Status\_ID] [int] not null,  constraint [PK\_Supplier\_Contracts] primary key clustered  ([ID\_Supplier\_Contracts] ASC) on [PRIMARY],  constraint [UQ\_Contract\_Number\_Supplier\_Contracts] unique ([Contract\_Number\_Supplier\_Contracts]),  constraint [CH\_Contract\_Number\_Supplier\_Contracts] check ([Contract\_Number\_Supplier\_Contracts] like 'Д-П/[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\_Date\_Of\_Formation\_Supplier\_Contracts] check ([Date\_Of\_Formation\_Supplier\_Contracts] <= getdate()),  constraint [CH\_Validity\_Period\_Supplier\_Contracts] check ([Validity\_Period\_Supplier\_Contracts] like '%[0-9]%'),  constraint [FK\_Supplier\_Supplier\_Contracts] foreign key ([Supplier\_ID])  references [dbo].[Suppliers] ([ID\_Supplier]),  constraint [FK\_Order\_Status\_Supplier\_Contracts] foreign key ([Order\_Status\_ID])  references [dbo].[Order\_Status] ([ID\_Order\_Status])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Supplier\_Contracts] (  [Supplier\_ID],  [Contract\_Number\_Supplier\_Contracts],  [Date\_Of\_Formation\_Supplier\_Contracts],  [Validity\_Period\_Supplier\_Contracts],  [Order\_Status\_ID]  )  values  (2, 'Д-П/0000000001/23', CONVERT(date, '2023-01-02', 120), '1 год', 2),  (2, 'Д-П/0000000002/23', CONVERT(date, '2023-04-17', 120), '1 год', 1),  (1, 'Д-П/0000000001/24', CONVERT(date, '2024-01-03', 120), '2 года', 1),  (2, 'Д-П/0000000002/24', CONVERT(date, '2024-02-10', 120), '1 год', 1);  go |
| Запросы на выборку данных |
| **Задача:** Вывести информацию о договорах поставщиков, отсортировав по дате формирования договора. |
| select  [Supplier\_ID] as "ID Поставщика",  [Contract\_Number\_Supplier\_Contracts] as "Номер договора",  [Date\_Of\_Formation\_Supplier\_Contracts] as "Дата формирования",  [Validity\_Period\_Supplier\_Contracts] as "Срок действия",  [Order\_Status\_ID] as "ID Статуса заказа"  from [dbo].[Supplier\_Contracts]  order by [Date\_Of\_Formation\_Supplier\_Contracts] ASC;  go |
| Результат запроса на выборку данных | |
|  | |
| Таблица | Изменение структуры таблицы |
| create table [dbo].[Inventory\_Supplier\_Contracts]  (  [ID\_Inventory\_Supplier\_Contracts] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Supplier\_Contracts\_ID] [int] not null,  [Count\_Inventory\_Supplier\_Contracts] [int] not null,  constraint [PK\_Inventory\_Supplier\_Contracts] primary key clustered  ([ID\_Inventory\_Supplier\_Contracts] ASC) on [PRIMARY],  constraint [CH\_Count\_Inventory\_Supplier\_Contracts] check ([Count\_Inventory\_Supplier\_Contracts] > 0),  constraint [FK\_Inventory\_Inventory\_Supplier\_Contracts] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory]),  constraint [FK\_Supplier\_Contracts\_Inventory\_Supplier\_Contracts] foreign key ([Supplier\_Contracts\_ID])  references [dbo].[Supplier\_Contracts] ([ID\_Supplier\_Contracts])  )  go | - |
| Запросы на добавление данных |
| insert into [dbo].[Inventory\_Supplier\_Contracts] (  [Inventory\_ID],  [Supplier\_Contracts\_ID],  [Count\_Inventory\_Supplier\_Contracts]  )  values  (1, 1, 7),  (3, 1, 5),  (7, 1, 40),  (1, 2, 5),  (6, 2, 25),  (8, 2, 10),  (6, 3, 25),  (8, 3, 15),  (3, 4, 5),  (7, 4, 10),  (9, 4, 30);  go |
| Запросы на выборку данных |
| **Задача:** Вывести информацию об инвентаре, связанном с договорами поставщиков, отсортированную по ID инвентаря. |
| select  [Inventory\_ID] as "ID Инвентаря",  [Supplier\_Contracts\_ID] as "ID Договора поставщика",  [Count\_Inventory\_Supplier\_Contracts] as "Количество инвентаря"  from [dbo].[Inventory\_Supplier\_Contracts]  order by [Inventory\_ID] ASC;  go |
| Результат запроса на выборку данных | |
|  | |

1. Изменение и удаление данных в таблицах

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Armor\_Type]  (  [ID\_Armor\_Type] [int] not null identity(1,1),  [Inventory\_ID] [int] not null,  [Name\_Armor\_Type] [varchar](15) not null,  constraint [PK\_Armor\_Type] primary key clustered  ([ID\_Armor\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Armor\_Type] unique ([Name\_Armor\_Type]),  constraint [FK\_Inventory\_Armor\_Type] foreign key ([Inventory\_ID])  references [dbo].[Inventory] ([ID\_Inventory])  )  go | Изменение данных |
| update [dbo].[Armor\_Type]  set [Name\_Armor\_Type] = 'Шлем - 1'  where [ID\_Armor\_Type] = 1;  go |
|  |
| Удаление данных |
| delete from [dbo].[Armor\_Type]  where [ID\_Armor\_Type] = 1;  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Mark]  (  [ID\_Mark] [int] not null identity(1,1),  [Name\_Mark] [varchar](15) not null,  constraint [PK\_Mark] primary key clustered  ([ID\_Mark] ASC) on [PRIMARY],  constraint [UQ\_Name\_Mark] unique ([Name\_Mark])  )  go | Изменение данных |
| update [dbo].[Mark]  set [Name\_Mark] = 'MarkerShot-1'  where [Name\_Mark] = 'MarkerShot';  go |
|  |
| Удаление данных |
| delete from [dbo].[Mark]  where [Name\_Mark] = 'MarkerShot-1';  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Model]  (  [ID\_Model] [int] not null identity(1,1),  [Name\_Model] [varchar](15) not null,  [Mark\_ID] [int] not null,  constraint [PK\_Model] primary key clustered  ([ID\_Model] ASC) on [PRIMARY],  constraint [UQ\_Name\_Model] unique ([Name\_Model]),  constraint [FK\_Mark\_Model] foreign key ([Mark\_ID])  references [dbo].[Mark] ([ID\_Mark])  )  go | Изменение данных |
| update [dbo].[Model]  set [Name\_Model] = 'Mrk I-1'  where [Name\_Model] = 'Mrk I';  go |
|  |
| Удаление данных |
| delete from [dbo].[Model]  where [Name\_Model] = 'Mrk I-1';  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Order\_Status]  (  [ID\_Order\_Status] [int] not null identity(1,1),  [Status\_Order\_Status] [varchar](8) not null,  constraint [PK\_Order\_Status] primary key clustered  ([ID\_Order\_Status] ASC) on [PRIMARY],  constraint [UQ\_Status\_Order\_Status] unique ([Status\_Order\_Status]),  constraint [CH\_Status\_Order\_Status] check ([Status\_Order\_Status] in ('открыт', 'завершён'))  )  go | Изменение данных |
| update [dbo].[Order\_Status]  set [Status\_Order\_Status] = 'Открыт-1'  where [Status\_Order\_Status] = 'Открыт';  go |
|  |
| Удаление данных |
| delete from [dbo].[Order\_Status]  where [Status\_Order\_Status] = 'Открыт';  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Player\_Type]  (  [ID\_Player\_Type] [int] not null identity(1,1),  [Name\_Player\_Type] [varchar](20) not null,  constraint [PK\_Player\_Type] primary key clustered  ([ID\_Player\_Type] ASC) on [PRIMARY],  constraint [UQ\_Name\_Player\_Type] unique ([Name\_Player\_Type]),  constraint [CH\_Name\_Player\_Type] check ([Name\_Player\_Type] in ('Ответственный', 'Участник'))  )  go | Изменение данных |
| update [dbo].[Player\_Type]  set [Name\_Player\_Type] = 'Участник-1'  where [Name\_Player\_Type] = 'Участник';  go |
|  |
| Удаление данных |
| delete from [dbo].[Player\_Type]  where [Name\_Player\_Type] = 'Участник';  go |
|  |

1. Фильтрация данных

|  |  |
| --- | --- |
| Таблица | Запросы и результаты |
| create table [dbo].[Contract]  (  [Employee\_ID] [int] not null,  [Game\_Type\_ID] [int] not null,  [Map\_ID] [int] not null,  [ID\_Contract] [int] not null identity(1,1),  [Number\_Contract] [varchar](10) not null,  [Cost\_Contract] [decimal](10,3) not null,  [DateTime\_Forming\_Contract] [datetime] not null,  [Game\_Duration\_Contract] [decimal](5,2) not null,  [Game\_Start\_Time\_Contract] [time] not null,  [Game\_End\_Time\_Contract] [time] not null,  [Final\_Cost\_Contract] [decimal](10,3) not null,  constraint [PK\_Contract] primary key clustered  ([ID\_Contract] ASC) on [PRIMARY],  constraint [UQ\_Number\_Contract] unique ([Number\_Contract]),  constraint [CH\_Number\_Contract] check ([Number\_Contract] like 'GM-[0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Cost\_Contract] check ([Cost\_Contract] >= 0),  constraint [CH\_Game\_Duration\_Contract] check ([Game\_Duration\_Contract] >= 0),  constraint [CH\_Game\_Start\_Time\_Contract] check ([Game\_Start\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Game\_End\_Time\_Contract] check ([Game\_End\_Time\_Contract] > [Game\_Start\_Time\_Contract] and [Game\_End\_Time\_Contract] > cast([DateTime\_Forming\_Contract] as time)),  constraint [CH\_Final\_Cost\_Contract] check ([Final\_Cost\_Contract] > 0),  constraint [FK\_Employee\_Contract] foreign key ([Employee\_ID])  references [dbo].[Employee] ([ID\_Employee]),  constraint [FK\_Game\_Type\_Contract] foreign key ([Game\_Type\_ID])  references [dbo].[Game\_Type] ([ID\_Game\_Type]),  constraint [FK\_Map\_Contract] foreign key ([Map\_ID])  references [dbo].[Map] ([ID\_Map])  )  go | Задача: Вывести информацию о договорах с клиентами, включая данные об ответственном сотруднике, тип игры, карту, и общую стоимость договора. |
| select  c.Number\_Contract as 'Номер Договора',  concat(e.Surname\_Employee, ' ', e.Name\_Employee, ' ', e.Patronymic\_Employee) as 'Ответственный Сотрудник',  g.Name\_Game\_Type as 'Тип Игры',  m.Name\_Map as 'Карта',  c.Cost\_Contract as 'Стоимость Договора',  c.Game\_Duration\_Contract as 'Длительность Игры (часы)',  c.Final\_Cost\_Contract as 'Итоговая Стоимость'  from  [dbo].[Contract] c  join [dbo].[Employee] e on c.Employee\_ID = e.ID\_Employee  join [dbo].[Game\_Type] g on c.Game\_Type\_ID = g.ID\_Game\_Type  join [dbo].[Map] m on c.Map\_ID = m.ID\_Map;  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Inventory]  (  [ID\_Inventory] [int] not null identity(1,1),  [Number\_Inventory] [varchar](16) not null,  [Count\_Inventory] [int] not null,  [Cost\_Inventory] [decimal](10,3) not null,  constraint [PK\_Inventory] primary key clustered  ([ID\_Inventory] ASC) on [PRIMARY],  constraint [UQ\_Number\_Inventory] unique ([Number\_Inventory]),  constraint [CH\_Number\_Inventory] check ([Number\_Inventory] like 'ИнВ-[0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]'),  constraint [CH\_Count\_Inventory] check ([Count\_Inventory] >= 0),  constraint [CH\_Cost\_Inventory] check ([Cost\_Inventory] >= 0)  )  go | Задача: Вывести информацию о инвентаре, который был задействован в определённых договорах, включая модель инвентаря и количество инвентаря в каждом договоре. |
| select  c.Number\_Contract as 'Номер Договора',  concat(m.Name\_Mark, ' ', mo.Name\_Model) as 'Модель Инвентаря',  i.Count\_Inventory as 'Количество Инвентаря'  from  [dbo].[Inventory\_Contract] ic  join [dbo].[Inventory] i on ic.Inventory\_ID = i.ID\_Inventory  join [dbo].[Mark\_Model\_Inventory] mm on i.ID\_Inventory = mm.Inventory\_ID  join [dbo].[Model] mo on mm.Model\_ID = mo.ID\_Model  join [dbo].[Mark] m on mo.Mark\_ID = m.ID\_Mark  join [dbo].[Contract] c on ic.Contract\_ID = c.ID\_Contract;  go |
|  |
| Таблица | Запросы и результаты |
| create table [dbo].[Client\_Contract]  (  [ID\_Client\_Contract] [int] not null identity(1,1),  [Contract\_ID] [int] not null,  [Client\_ID] [int] not null,  [Player\_Type\_ID] [int] not null,  constraint [PK\_Client\_Contract] primary key clustered  ([ID\_Client\_Contract] ASC) on [PRIMARY],  constraint [FK\_Contract\_Client\_Contract] foreign key ([Contract\_ID])  references [dbo].[Contract] ([ID\_Contract]),  constraint [FK\_Client\_Client\_Contract] foreign key ([Client\_ID])  references [dbo].[Client] ([ID\_Client]),  constraint [FK\_Player\_Type\_Client\_Contract] foreign key ([Player\_Type\_ID])  references [dbo].[Player\_Type] ([ID\_Player\_Type])  )  go | Задача: Вывести список клиентов и их участие в договорах, указав тип игрока (участник или ответственный), и информацию о связанных с ними дополнительных контрактах. |
| select  concat(cl.Surname\_Client, ' ', cl.Name\_Client, ' ', cl.Patronymic\_Client) as 'Клиент',  acc.Number\_Additional\_Contract as 'Номер Дополнительного Договора',  pt.Name\_Player\_Type as 'Тип Игрока',  co.Number\_Contract as 'Номер Основного Договора'  from  [dbo].[Client\_Contract] cc  join [dbo].[Client] cl on cc.Client\_ID = cl.ID\_Client  join [dbo].[Player\_Type] pt on cc.Player\_Type\_ID = pt.ID\_Player\_Type  join [dbo].[Contract] co on cc.Contract\_ID = co.ID\_Contract  left join [dbo].[Additional\_Client\_Contract] acc on cc.ID\_Client\_Contract = acc.Client\_Contract\_ID;  go |
|  |