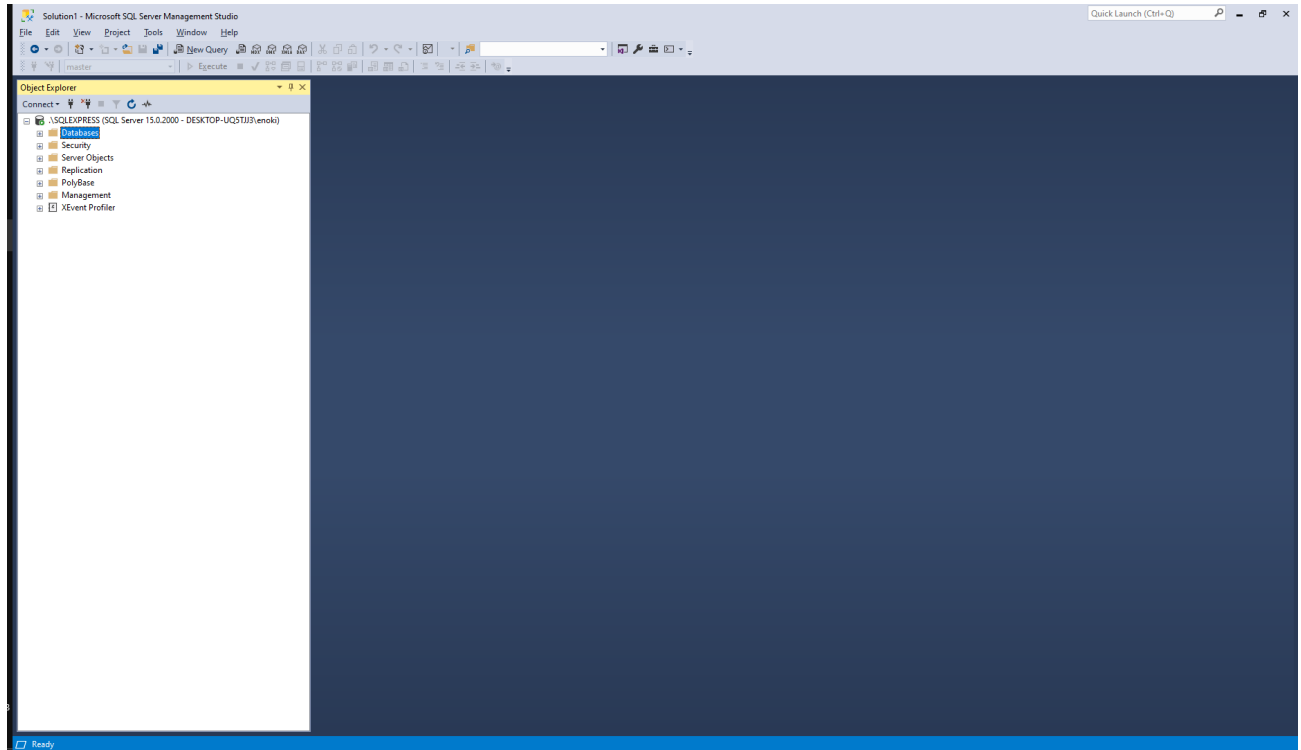


Всем снова привет!

Сегодня мы создадим свою базу данных, на которой потом будем обучаться работать, изменять таблицы, строить возможно схемы, и нормализовывать данные.

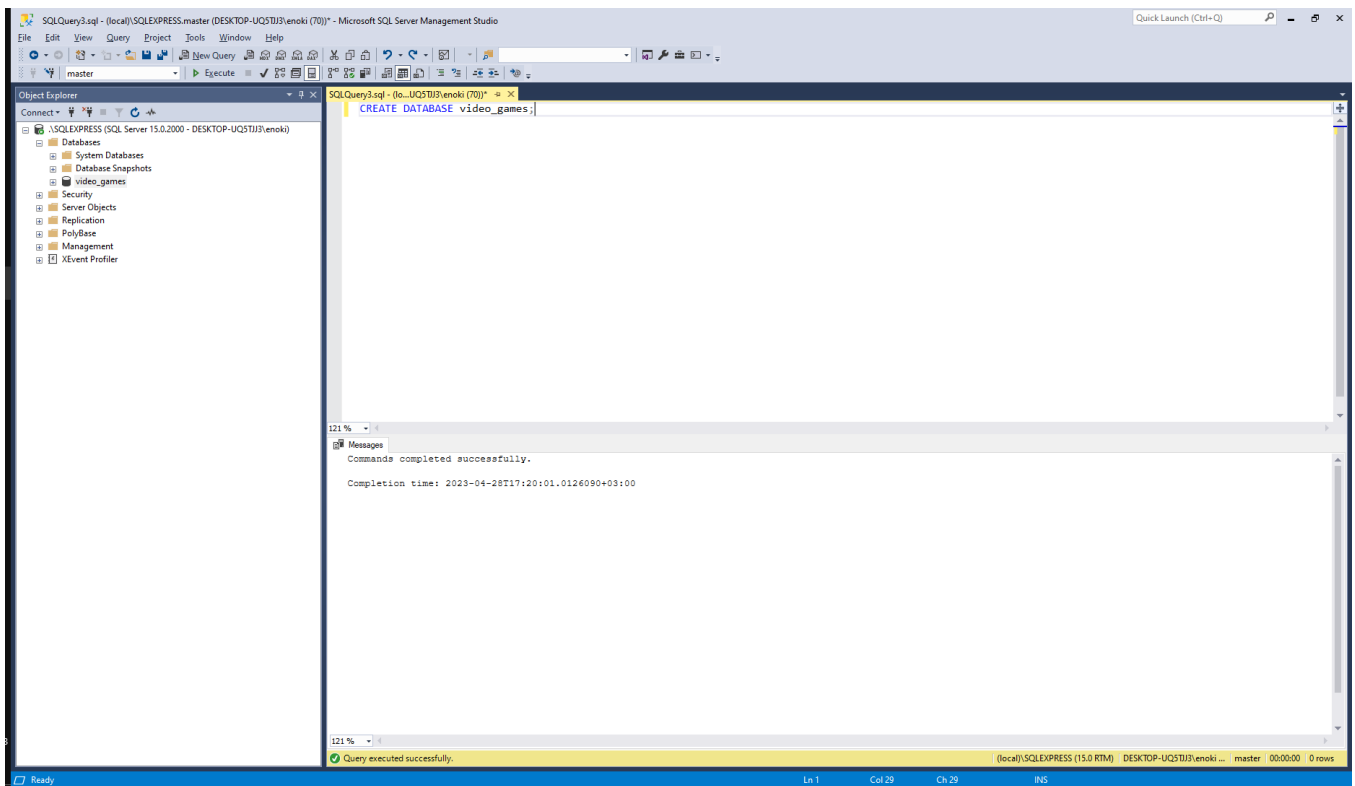
Поехали!

1. Запустим Microsoft SSMS



2. Создадим командой новую БД, назовем ее video_games:

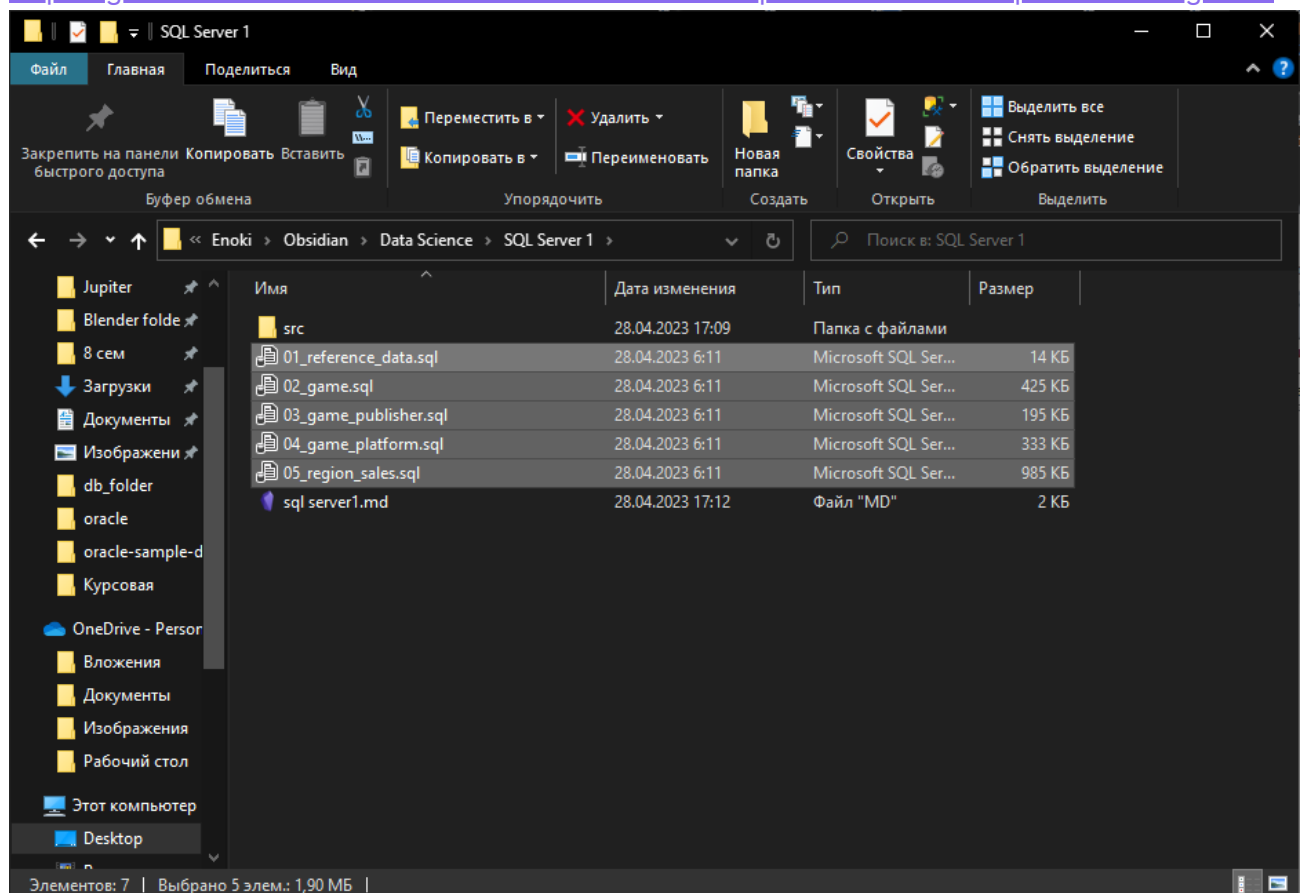
```
CREATE DATABASE video_games;
```



3. Скачиваем уже готовые таблицы из интернета (для обучения в самый раз)

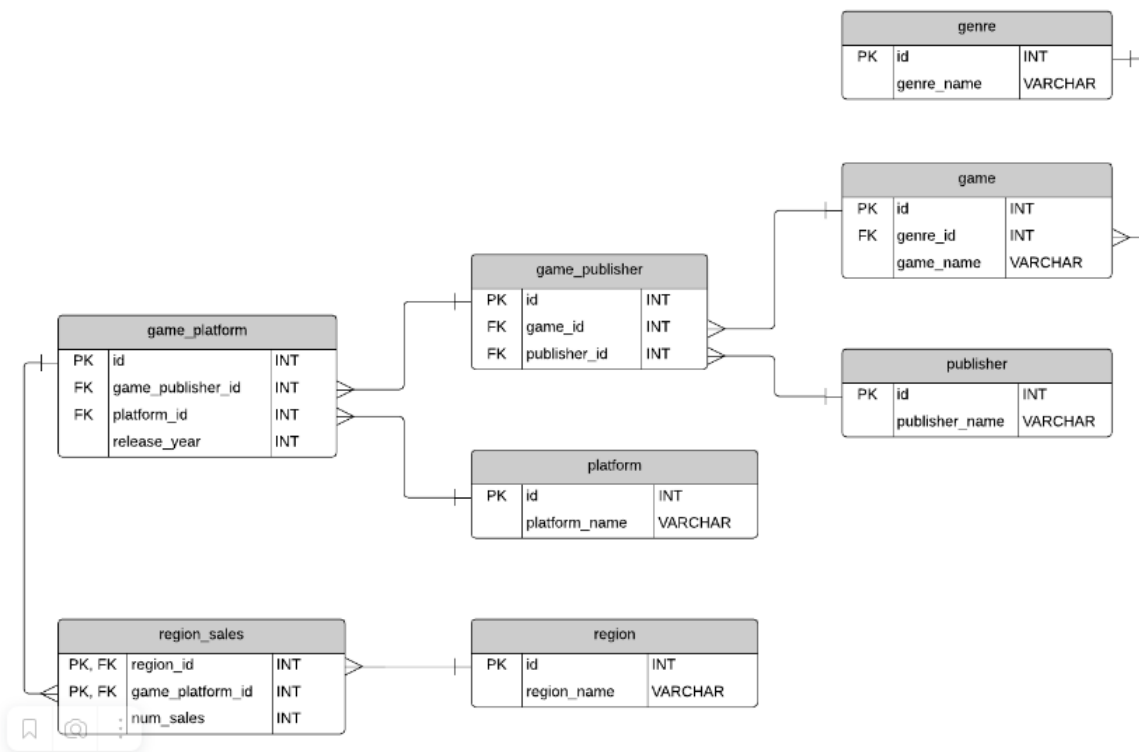
Сделаем это по готовым примерам таблиц, скачать можно, к примеру, по ссылке

https://github.com/bbrumm/databasestar/tree/main/sample_databases/sample_db_videogames



4. Поочередно открываем их на сервере, и выполняем программы по заполнению новых таблиц

5. Рассмотрим схему всех таблиц:



Посмотрим на парочку из них и убедимся, что все они составлены именно так (вдруг обманывают)

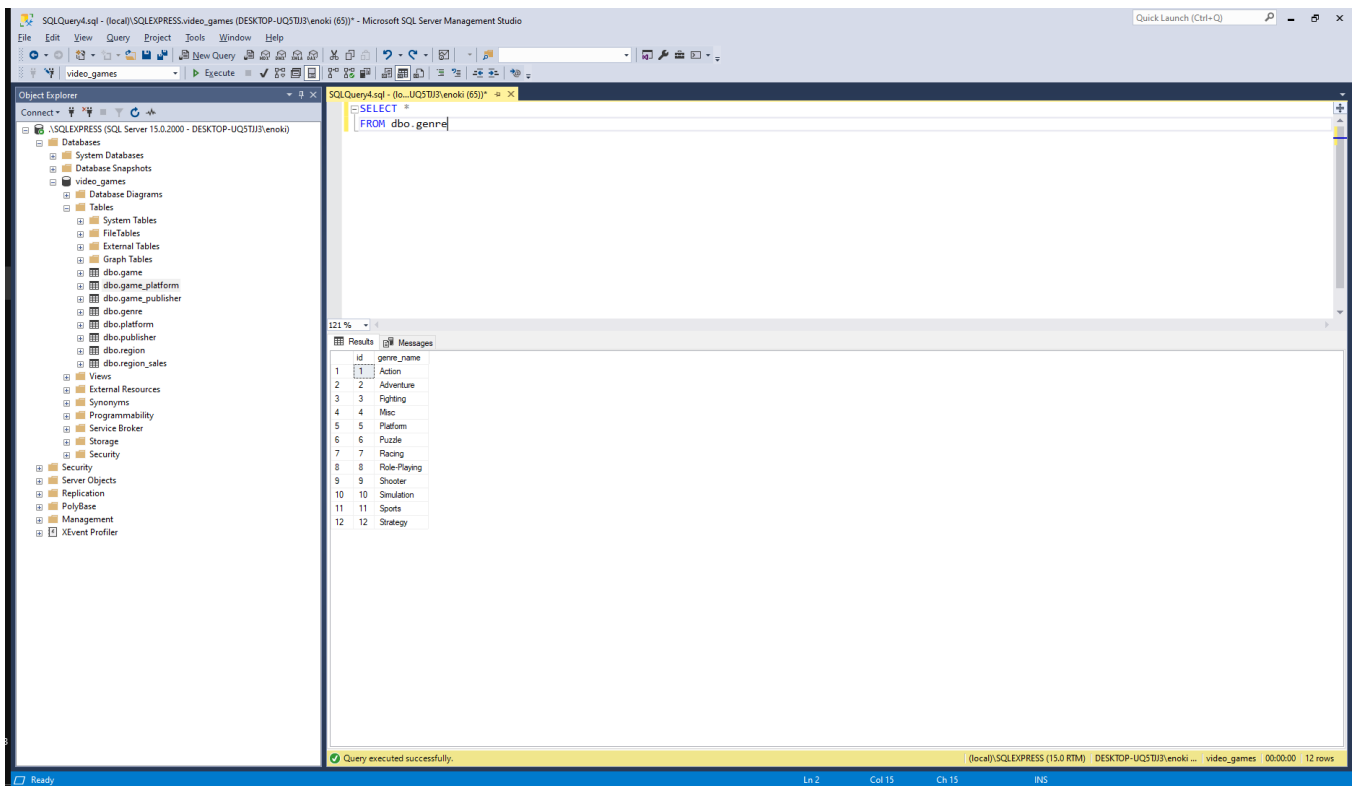
The screenshot shows the Microsoft SQL Server Management Studio interface. The query window displays the following SQL query:

```
SELECT *
FROM dbo.game_platform
```

The Results pane shows the following data:

	id	game_publisher_id	platform_id	release_year
1	6564	4	7	2007
2	9852	4	7	2007
3	11063	7	15	2006
4	9065	15	15	2011
5	9544	15	15	2011
6	11623	15	15	2009
7	1977	16	16	1981
8	3833	16	16	1981
9	3838	16	16	1982
10	4755	16	16	1982
11	5034	16	16	1981
12	9762	15	15	2008
13	3647	7	7	2002
14	3657	7	7	2003
15	5747	7	7	2001
16	5788	7	7	2000
17	6318	7	7	2001
18	6544	7	7	2001
19	6657	7	7	2001
20	7051	7	7	2002
21	7172	7	7	2000
22	8175	7	7	2002
23	8750	7	7	2001
24	9032	7	7	2002
25	9522	7	7	2001
26	1952	12	12	1999
27	3341	12	12	2000
28	3532	12	12	1999

The status bar at the bottom indicates: Query executed successfully. (local) \SQL\EXPRESS (15.0 RTM) DESKTOP-UQSTJ3J\enoki ... video_games 00:00:00 16 326 rows



Всё сходится! Можем начинать строить себе задачи, которые могут давать нам работодатели

Задание №1.

Создайте новое представление, которое будет показывать только игры для ПК платформ, выпущенных до 2002 года

Комментарий:

Мда уж, сам себе поставил непростую задачу.

Давайте обратимся к заранее изученным познаниям того, что такое представление, и в особенности, как его использовать в T-SQL (если кто не понял пока мы работаем в проге от Microsoft тут своя СУБД, но разницы особой нет).

Во время подготовки к одному из собеседований, которое так и не состоялось, я выписал следующие вещи:

Представление представляет собой хранимый запрос к базе данных, также ее можно назвать виртуальная таблица, но в этой таблице данные не хранятся, а хранится только сам запрос. Но, тем не менее, к вышке можно обращаться как к обычной таблице и извлекать данные из нее.

Например, в [PostgreSQL](#) запрос создания представления будет выглядеть так:

```
CREATE VIEW MyView
AS
    SELECT id, name, org
    FROM work.TableName
```

Полный синтаксис выйдет следующим образом:

```
CREATE [OR REPLACE]
[ALGORITHM = {UNDEFINED | MERGE | TEMPTABLE}]
VIEW view_name [(column_list)]
```

```
AS select_statement  
[WITH [CASCADED | LOCAL] CHECK OPTION]
```

Попробуем для начала реализовать запрос с выводом топа, а потом уже будем записывать его в представление, хорошо? Надеюсь, что да.

Решение

Если немного покопаться, то можно прийти к такой команде:

```
SELECT g.game_name Игра, gp.release_year Год_выпуска, p.platform_name Платформа  
FROM dbo.game g  
JOIN dbo.game_platform gp  
ON g.id=gp.id  
JOIN dbo.platform p  
ON gp.platform_id=p.id  
WHERE p.platform_name='PC'  
AND gp.release_year<2002
```

The screenshot shows the Microsoft SQL Server Management Studio interface. The query editor contains the following SQL query:

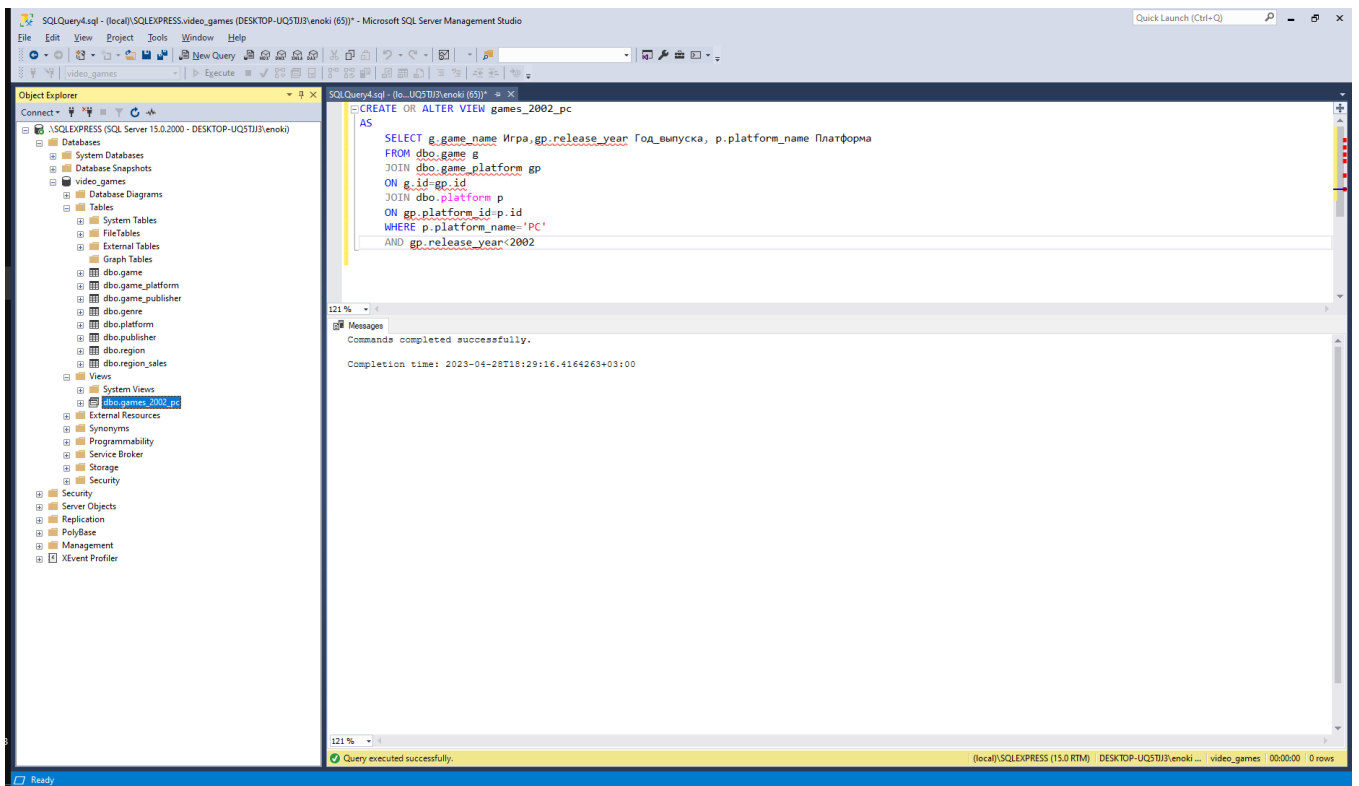
```
SELECT g.game_name Игра, gp.release_year Год_выпуска, p.platform_name Платформа  
FROM dbo.game g  
JOIN dbo.game_platform gp  
ON g.id=gp.id  
JOIN dbo.platform p  
ON gp.platform_id=p.id  
WHERE p.platform_name='PC'  
AND gp.release_year<2002
```

The query results are displayed in the Results pane, showing a list of games and their release years. The results are as follows:

Игра	Год_выпуска	Платформа
Assassin's Creed III: Liberation	1994	PC
Cata West: The Vanishing Files	1995	PC
Cal Damage	1994	PC
Centipede: Infestation	1996	PC
Chaos/Head - Love Chu/Chu!	1998	PC
Chaotic: Shadow Warriors	1985	PC
Dragon Shadow Spell	2001	PC
Dragon Slayer: The Legend of Heroes	2001	PC
Hakuoki: Remastered DS	1999	PC
Hashou Pachinko"Pachi-Slot Kouyaku Series Vol. ...	1997	PC
Hashou Pachinko"Pachi-Slot Kouyaku Series Vol. ...	1996	PC
History Great Empires: Rome	1998	PC
Hitman 2: Silent Assassin	1998	PC
Hitman: Blood Money	1999	PC
Homeworld: The Revolution	2001	PC
Home Rollerz	2000	PC
Let's Make a Soccer Team!	2001	PC
Let's Sing 2016	2000	PC
Lethal Skies II	2000	PC
Lio & Stitch 2: Hamsterniel Havoc	1997	PC
Line Rider 2: Unbound	2001	PC
Lips	1998	PC
Little Battles eXperience: Baku Boost	2001	PC
Little League World Series Baseball: Double Play	2000	PC
Loons: The Fight For Fame	1992	PC
Lost in Shadow	1999	PC
Lost Odyssey	2001	PC
Marvel: Ultimate Alliance 2	1999	PC
MiniCopter: Adventure Flight	2001	PC
MotionSports: Adrenaline	1994	PC
Moto Racer	2000	PC
My Fitness Coach: Club	1992	PC
Need for Speed Carbon	1998	PC

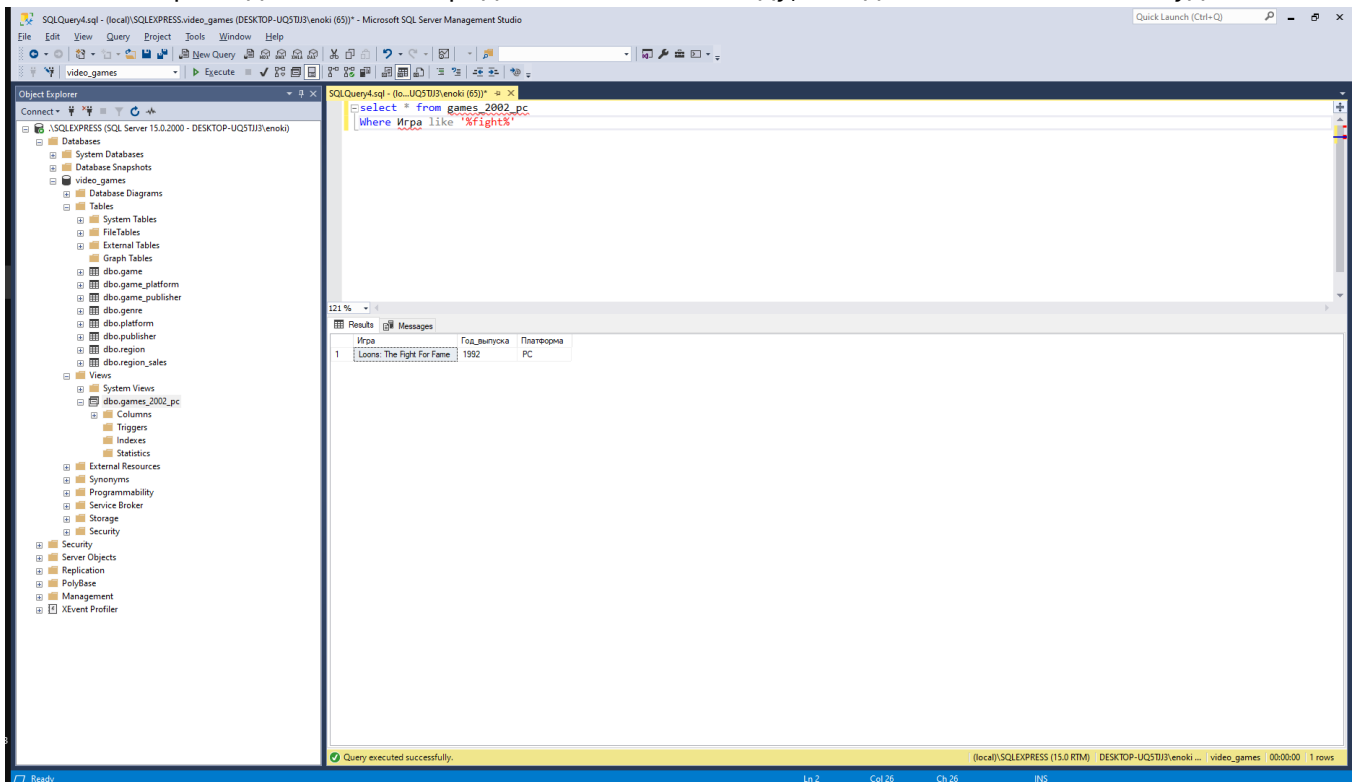
Осталось только создать представление.

```
CREATE OR ALTER VIEW games_2002_pc  
AS  
SELECT g.game_name Игра, gp.release_year Год_выпуска, p.platform_name Платформа  
FROM dbo.game g  
JOIN dbo.game_platform gp  
ON g.id=gp.id  
JOIN dbo.platform p  
ON gp.platform_id=p.id  
WHERE p.platform_name='PC'  
AND gp.release_year<2002
```



Готово! Теперь у нас есть представление, которое можно использовать в своих нуждах.

Можно теперь подставить это представление в команду, и поделаться с этим что-нибудь.



Задача 2.

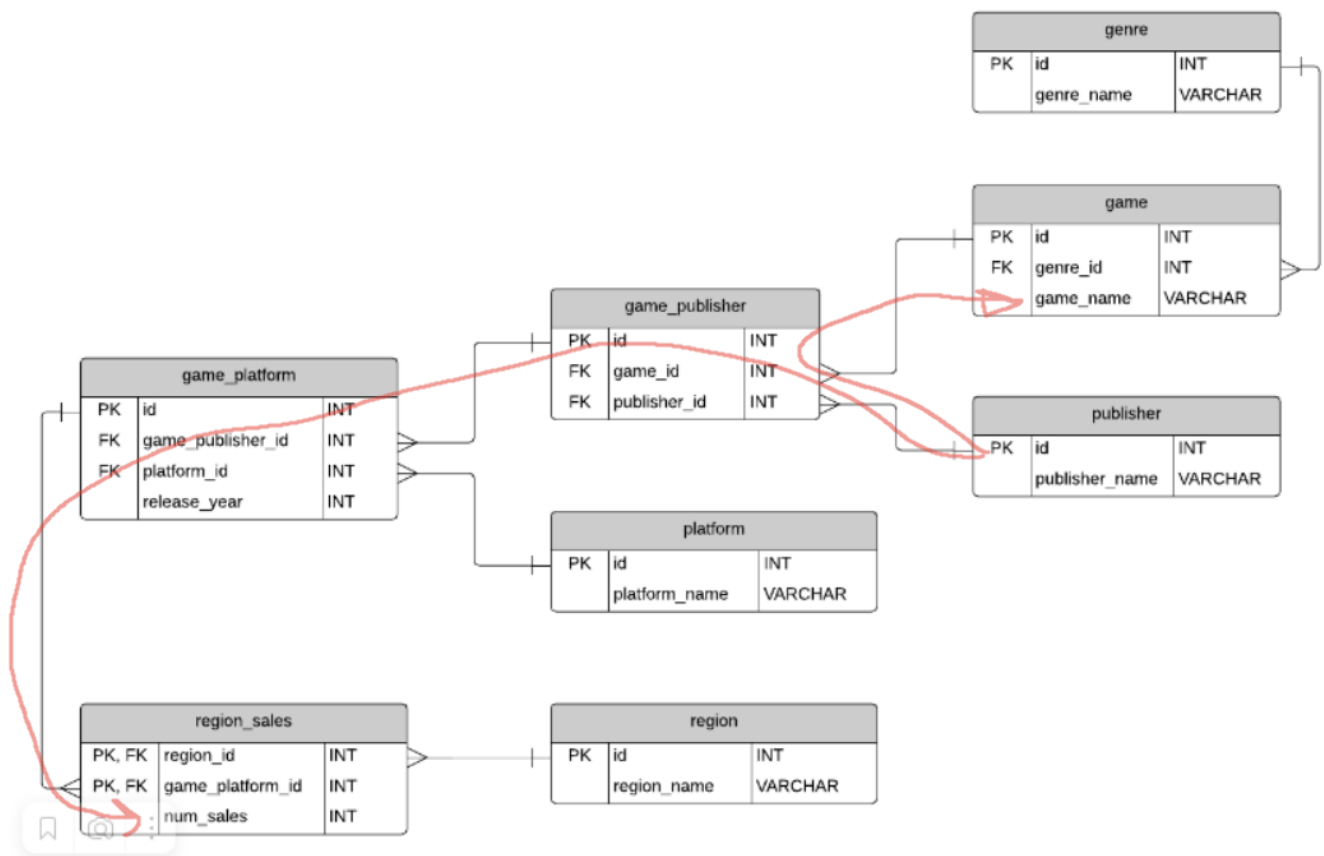
Выведите самые продаваемые игры для каждой компании в БД.

Таблица должна выглядеть:

publisher_name, best_game, global_sales.

Решение:

Давайте посмотрим на схеме, как нам решать эту задачу:



Решение:

Делал я это почему-то долго, но написал в итоге только через WITH.

```
WITH task2
AS
    (SELECT game_name,publisher_name,num_sales
    FROM dbo.publisher pub
    JOIN dbo.game_publisher gpub ON gpub.publisher_id=pub.id
    JOIN dbo.game_platform gpl ON gpl.game_publisher_id=gpub.id
    JOIN dbo.region_sales rsales ON rsales.game_platform_id=gpl.id
    JOIN dbo.game game ON game.id=gpub.game_id
    ),
task2_2
AS
    (SELECT game_name best_game, SUM(num_sales) global_sales
    FROM task2
    GROUP BY game_name),
t3
AS (SELECT publisher_name,MAX(global_sales) best_sale
FROM task2_2 t2
JOIN task2 t ON t.game_name=t2.best_game
GROUP BY publisher_name)

SELECT DISTINCT t3.publisher_name, t2.best_game, t3.best_sale
FROM task2
JOIN task2_2 t2 ON t2.best_game=task2.game_name
JOIN t3 ON task2.publisher_name=t3.publisher_name
WHERE t2.global_sales=t3.best_sale
ORDER BY t3.best_sale DESC
```

SQLQuery1.sql - (local)\SQLEXPRESS\video_games (DESKTOP-UQ2TJH3\enoki) (211) - Microsoft SQL Server Enterprise

File Edit View Query Project Tools Window Help

video_games Execute

Object Explorer

Connect

- SQLSERVER (SQL Server 15.0.2000 - DESKTOP-UQ2TJH3\enoki)
 - Databases
 - System Databases
 - Database Snapshots
 - video_games
 - Database Diagrams
 - Tables
 - System Tables
 - FileTables
 - External Tables
 - Graph Tables
 - dbo.game
 - dbo.game_platform
 - dbo.game_publisher
 - dbo.genre
 - dbo.platform
 - dbo.publisher
 - dbo.region
 - dbo.region_sales
 - Views
 - System Views
 - dbo.games_2002_pc
 - External Resources
 - Synonyms
 - Programmability
 - Service Broker
 - Storage
 - Security
 - Server Objects
 - Replication
 - PolyBase
 - Management
 - XEEvent Profiler

SQLQuery1.sql - (local)\SQLEXPRESS\enoki (211) *

```

FROM dbo.publisher pub
JOIN dbo.game_publisher gpub ON gpub.publisher_id=pub.id
JOIN dbo.game_platform gpl ON gpl.publisher_id=gpub.id
JOIN dbo.region_sales rsales ON rsales.game_platform_id=gpl.id
JOIN dbo.game game ON game.id=gpub.game_id
),
task2_2
AS
(SELECT game_name best_game, SUM(num_sales) global_sales
FROM task2
GROUP BY game_name),
t3
AS (SELECT publisher_name, MAX(global_sales) best_sale
FROM task2_2 t2
JOIN task2 t ON t.game_name=t2.best_game
GROUP BY publisher_name)

SELECT DISTINCT t3.publisher_name, t2.best_game, t3.best_sale
FROM task2
JOIN task2_2 t2 ON t2.best_game=task2.game_name
JOIN t3 ON task2.publisher_name=t3.publisher_name
WHERE t2.global_sales=t3.best_sale
ORDER BY t3.best_sale DESC
  
```

121 %

Results Messages

	publisher_name	best_game	best_sale
1	Nintendo	Wii Sports	82.74
2	Take-Two Interactive	Grand Theft Auto V	55.92
3	Activision	Call of Duty: Modern Warfare 3	30.84
4	Microsoft Game Studios	Minecraft	23.75
5	Sony Computer Entertainment	Minecraft	23.75
6	Sony Computer Entertainment Europe	Minecraft	23.75
7	Bethesda Softworks	The Elder Scrolls V: Skyrim	19.28
8	Electronic Arts	FIFA 15	19.02
9	Aspyr	Guitar Hero III: Legends of Rock	16.42
10	LucasArts	LEGO Star Wars: The Complete Saga	15.73
11	Ubisoft	Assassin's Creed IV: Black Flag	13.20
12	Sega	Mario & Sonic at the Olympic Games	13.19
13	Warner Bros. Interactive Entertainment	Batman: Arkham City	11.18
14	505 Games	Zumba Fitness	9.79
15	Atari	Pac-Man	9.03
16	Namco Bandai Games	Pac-Man	9.03
17	Capcom	Resident Evil 5	8.69
18	Konami Digital Entertainment	Pro Evolution Soccer 2008	8.47

Query executed successfully.

(local)\SQLEXPRESS (15.0 KTM) | DESKTOP-UQ2TJH3\enoki | video_games | 00:00:02 | 597 rows

Ready Col 1 INS