Florea Adrian 342C3

# Proiect Baze de Date 2

- Descrierea bazei de date
  Aplicatia se ocupa cu reprezentarea unei platforme online de promovare a jocurilor PC, unde se pot vizualiza aspecte precum pretul, data lansarii, rating, dar si un magazin de unde se poate achizitiona produsul. Aplicatia a fost creata folosind MSSQL pe docker, MSSMS (Microsoft SQL Server Management Studio), un framework de Web Development folosind .NET cu EF Core.
  - Structura bazei de date
    Pentru crearea diagramei bazei de date am folosit platforma online sqldbm.com, impreuna cu Microsoft SQL Diagram maker:

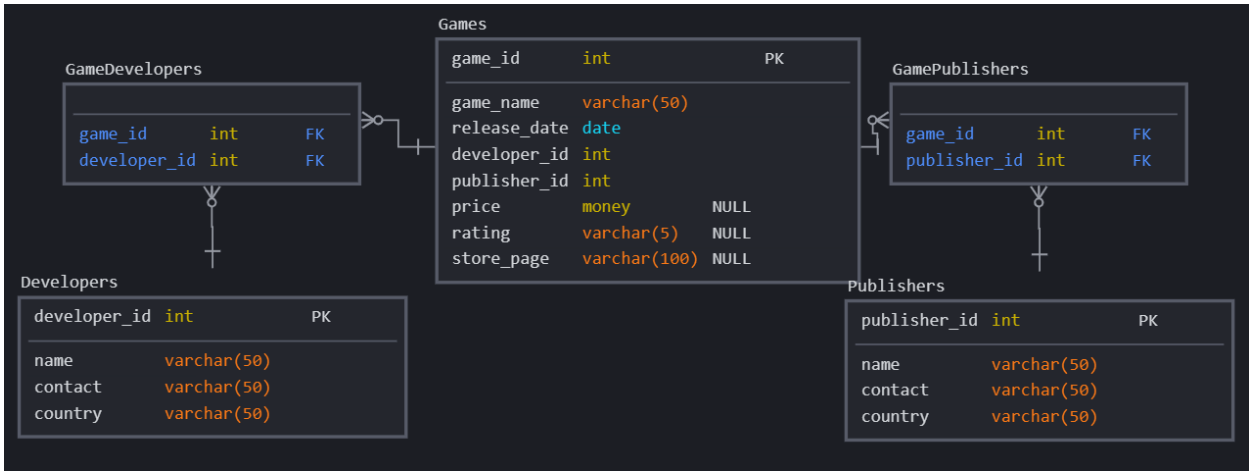


Fig. 1 Diagram sqldbm

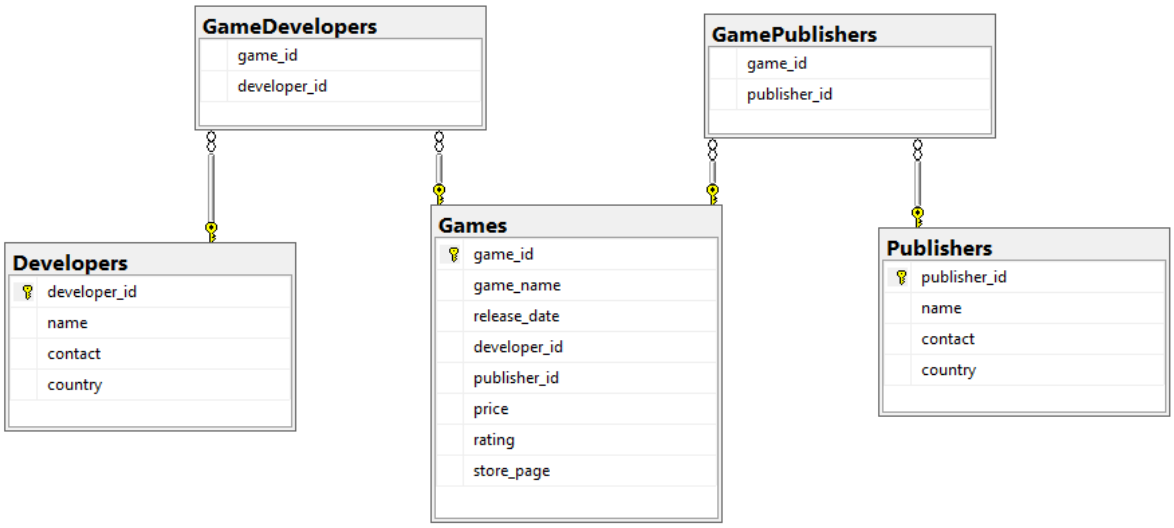


Fig. 2 Diagrama SSMS

- Descrierea constrângerilor de integritate
  In primul rand, principalele constrangeri de integritate reprezinta cheile primare din cele 3 tabele „Games”, „Developers” si „Publishers”, apoi secundare sunt foreign keys in tabelele „GameDevelopers” si „GamePublishers”. In plus, exista constrangeri legate de chei pentru ca acestea sa nu fie nule (NOT NULL).

```
USE [proiect]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Games](
    [game_id] [int] NOT NULL,
    [game_name] [varchar](50) NOT NULL,
    [release_date] [date] NOT NULL,
    [developer_id] [int] NOT NULL,
    [publisher_id] [int] NOT NULL,
    [price] [int] NOT NULL,
    [rating] [int] NULL,
 CONSTRAINT [PK_Games] PRIMARY KEY CLUSTERED
(
    [game_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Fig. 3 Crearea Tabelei Games**

```
USE [proiect]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Developers](
    [developer_id] [int] NOT NULL,
    [name] [varchar](50) NOT NULL,
    [contact] [varchar](50) NULL,
    [country] [varchar](50) NOT NULL,
 CONSTRAINT [PK_Developers] PRIMARY KEY CLUSTERED
(
    [developer_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Fig. 4 Crearea Tabelei Developers**

```
USE [proiect]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[Publishers](
    [publisher_id] [int] NOT NULL,
    [name] [varchar](50) NOT NULL,
    [contact] [varchar](50) NULL,
    [country] [varchar](50) NOT NULL,
 CONSTRAINT [PK_Publishers] PRIMARY KEY CLUSTERED
(
    [publisher_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
```

**Fig. 5 Crearea Tabelei Publishers**

```
USE [proiect]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[GameDevelopers](
    [game_id] [int] NOT NULL,
    [developer_id] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[GameDevelopers]  WITH CHECK ADD  CONSTRAINT [FK_GameDevelopers_Developers] FOREIGN KEY([developer_id])
REFERENCES [dbo].[Developers] ([developer_id])
GO

ALTER TABLE [dbo].[GameDevelopers] CHECK CONSTRAINT [FK_GameDevelopers_Developers]
GO

ALTER TABLE [dbo].[GameDevelopers]  WITH CHECK ADD  CONSTRAINT [FK_GameDevelopers_Games] FOREIGN KEY([game_id])
REFERENCES [dbo].[Games] ([game_id])
GO

ALTER TABLE [dbo].[GameDevelopers] CHECK CONSTRAINT [FK_GameDevelopers_Games]
GO
```

**Fig. 6 Crearea Tabelei GameDevelopers**

```
USE [proiect]
GO

SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[GamePublishers](
    [game_id] [int] NOT NULL,
    [publisher_id] [int] NOT NULL
) ON [PRIMARY]
GO

ALTER TABLE [dbo].[GamePublishers]  WITH CHECK ADD  CONSTRAINT [FK_GamePublishers_Games] FOREIGN KEY([game_id])
REFERENCES [dbo].[Games] ([game_id])
GO

ALTER TABLE [dbo].[GamePublishers] CHECK CONSTRAINT [FK_GamePublishers_Games]
GO

ALTER TABLE [dbo].[GamePublishers]  WITH CHECK ADD  CONSTRAINT [FK_GamePublishers_Publishers] FOREIGN KEY([publisher_id])
REFERENCES [dbo].[Publishers] ([publisher_id])
GO

ALTER TABLE [dbo].[GamePublishers] CHECK CONSTRAINT [FK_GamePublishers_Publishers]
GO
```

**Fig. 7 Crearea Tabelei GamePublishers**

- Descrierea procedurilor și funcțiilor
  In aplicatia creata am folosit o serie de proceduri: un trigger pentru a actualiza tabelele GameDevelopers si GamePublishers atunci cand tabela Games este actualizata, o procedura stocata pentru a primi toate produsele unei firme (toate jocurile publicate de un publisher), dar si alte proceduri stocate, de exemplu, pentru a gasi un developer sau un publisher dupa id-ul sau. In plus, am creat si o functie pentru a gasi jocurile mai ieftine de un anume prag.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE FUNCTION GetGamesLowerThan
(
    -- Add the parameters for the function here
    @Price int
)
RETURNS TABLE
AS
RETURN
(
    -- Add the SELECT statement with parameter references here
    SELECT *
    FROM [dbo].[Games]
    WHERE [price] < @Price
)
GO
```

Fig. 8 Crearea functiei

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =============================================
CREATE PROCEDURE GetGamesByPublisher
    -- Add the parameters for the stored procedure here
    -- @Publisher int
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    SELECT [game_id]
      ,[game_name]
      ,[release_date]
      ,[developer_id]
      ,[publisher_id]
      ,[price]
      ,[rating]
      ,[store_page]
  FROM [dbo].[Games]
  ORDER BY [publisher_id]
END
GO
```

Fig. 9 Crearea Procedurii Stocate

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =============================================
-- Author:      <Florea Adrian>
-- Description: <ProiectBD2>
-- =============================================
CREATE TRIGGER GamesTrigger
   ON  [dbo].[Games]
   AFTER INSERT
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for trigger here
    INSERT INTO [dbo].[GameDevelopers] (game_id)
    SELECT game_id
           FROM [dbo].[Games]
           WHERE game_id IS NOT NULL

    INSERT INTO [dbo].[GameDevelopers] (developer_id)
    SELECT developer_id
           FROM [dbo].[Developers]
           WHERE developer_id IS NOT NULL

    INSERT INTO [dbo].[GamePublishers] (game_id)
    SELECT game_id
           FROM [dbo].[Games]
           WHERE game_id IS NOT NULL

    INSERT INTO [dbo].[GamePublishers] (publisher_id)
    SELECT publisher_id
           FROM [dbo].[Publishers]
           WHERE publisher_id IS NOT NULL
END
GO
```

**Fig. 10 Crearea Trigger-ului**

- Descrierea aplicatiei
  Cum am mentionat si in introducere, aplicatia face legatura cu baza de date stocata pe docker folosind un framework de la microsoft bazat pe .NET, folosind Visual Studio Community 19:

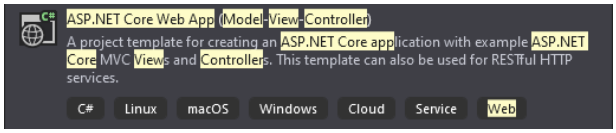


**Fig. 11 Visual Studio Proiect**

Folosind EF Core si .NET, ne putem conecta la baza de date, si astfel framework-ul creaza o serie de clase similare cu structura bazei de date, dar si tine cont de informatiile stocate in baza de date.

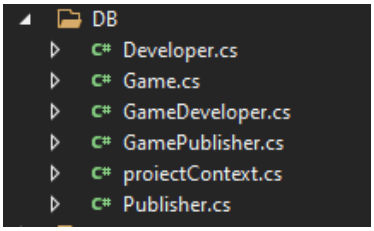


**Fig. 12 Ierarhie Clase Preluate din BD**

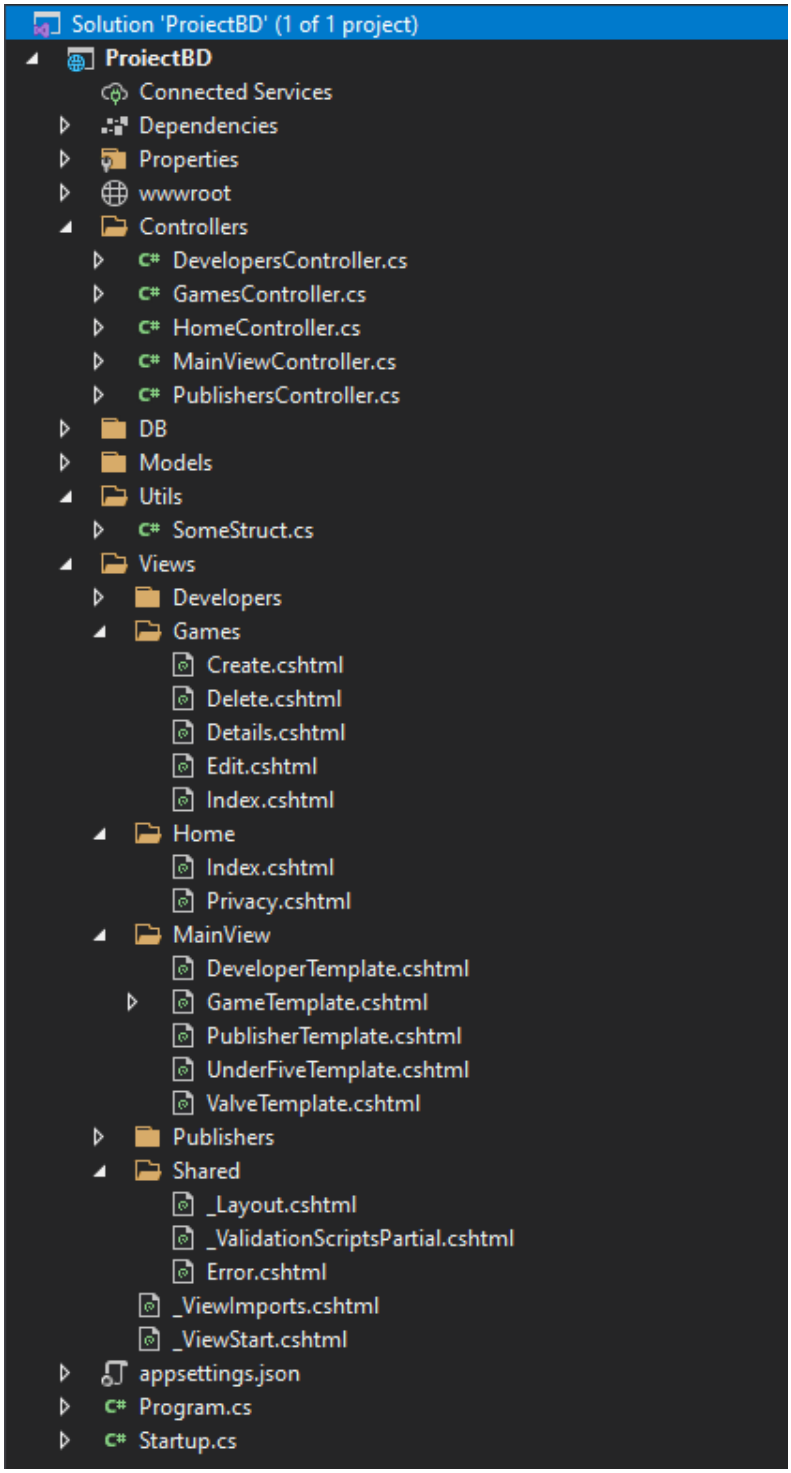


Fig. 13 Ierarhie Clase Proiect

- Conectarea la baza de date

Pentru a ma conecta la baza de date folosind framework-ul creat, am folosit urmatoarea comanda in powershell:

```
PS D:\Adi\BD2\proiect\vs\ProiectBD> docker start proiectBD
                                   proiectBD
PS D:\Adi\BD2\proiect\vs\ProiectBD> dotnet ef dbcontext scaffold "Data Source=localhost,1433;Initial Catalog=proiect;Persist Security
 Info=True;User ID=SA;Password=parolaAiaPuternic4!" Microsoft.EntityFrameworkCore.SqlServer -o DB --force
```

**Fig. 14 Conectare Aplicatie BD**

Aceasta comanda creaza folder-ul mentionat anterior pe nume „DB” (de aici parametrul –o DB –force).

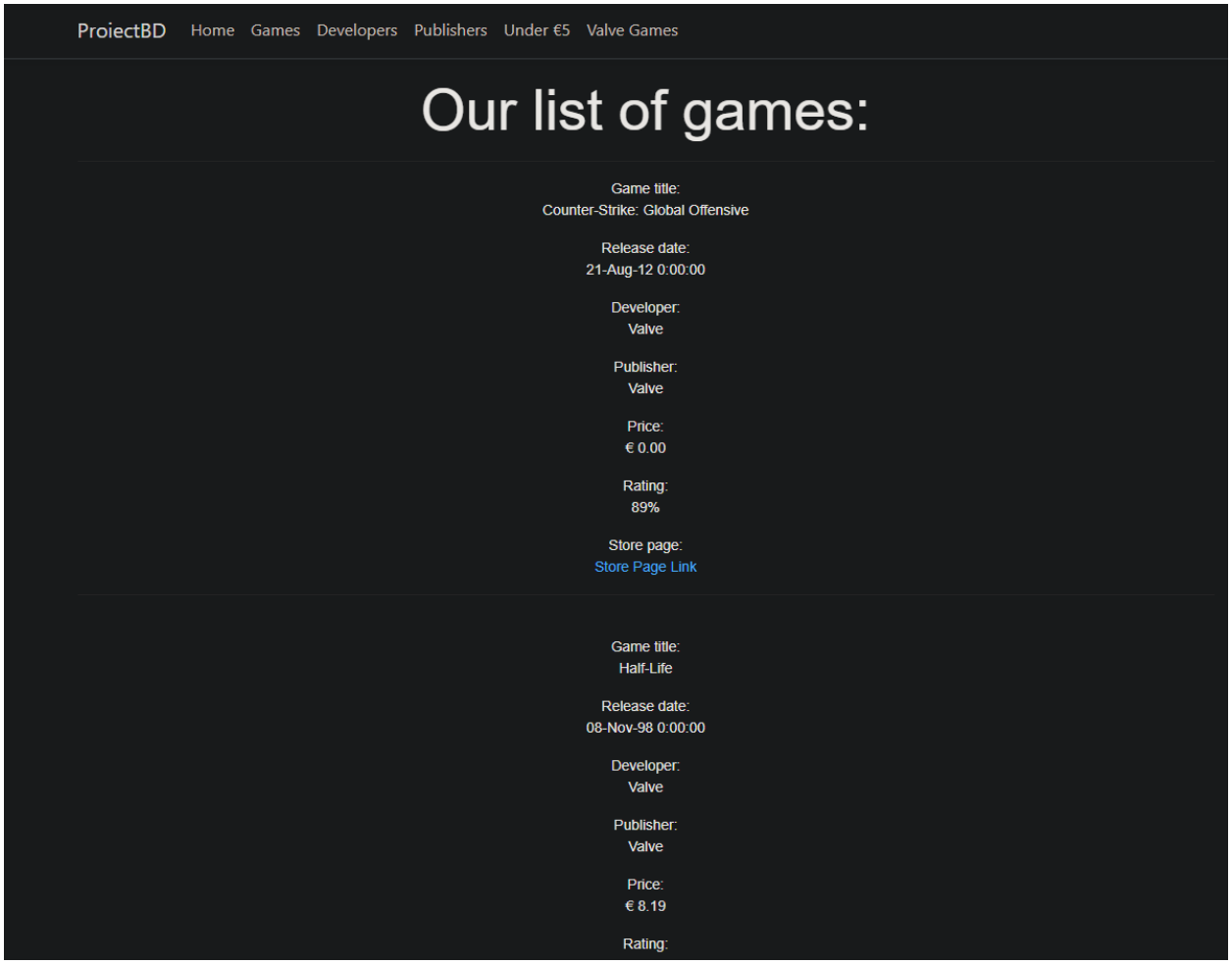


**Fig. 15 Interfata 1**

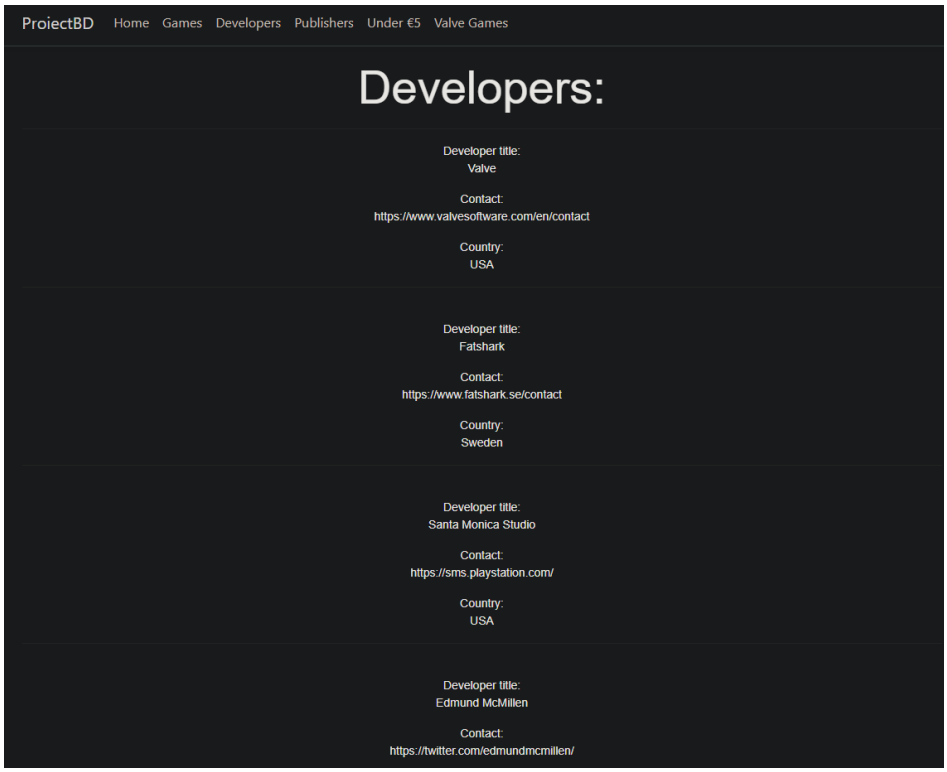


Fig. 16 Interfata 2

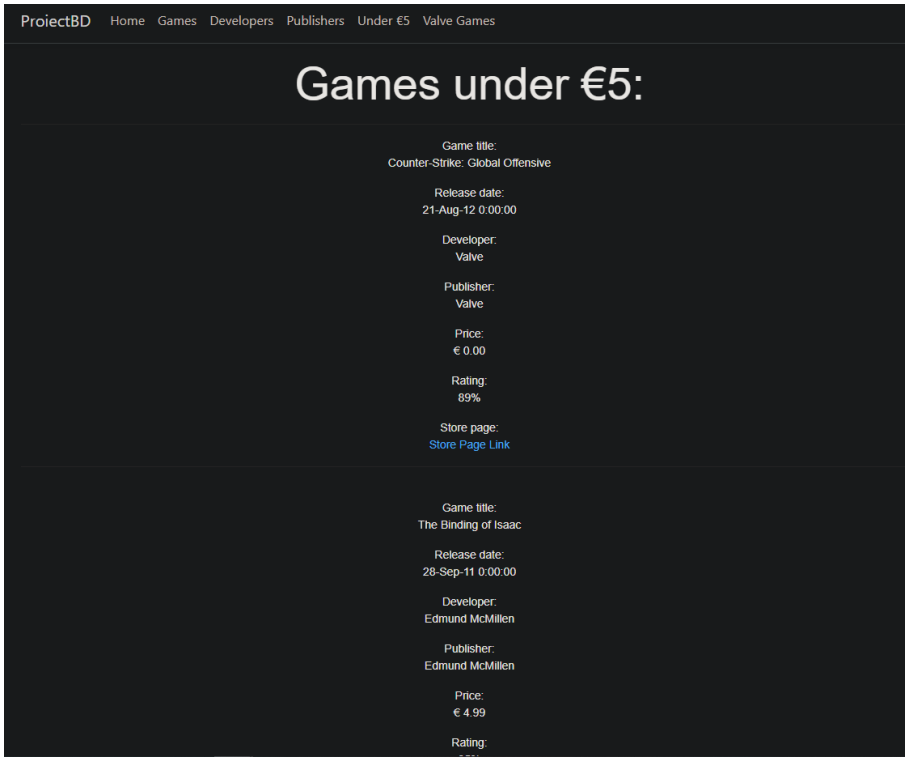


Fig. 17 Raport 1

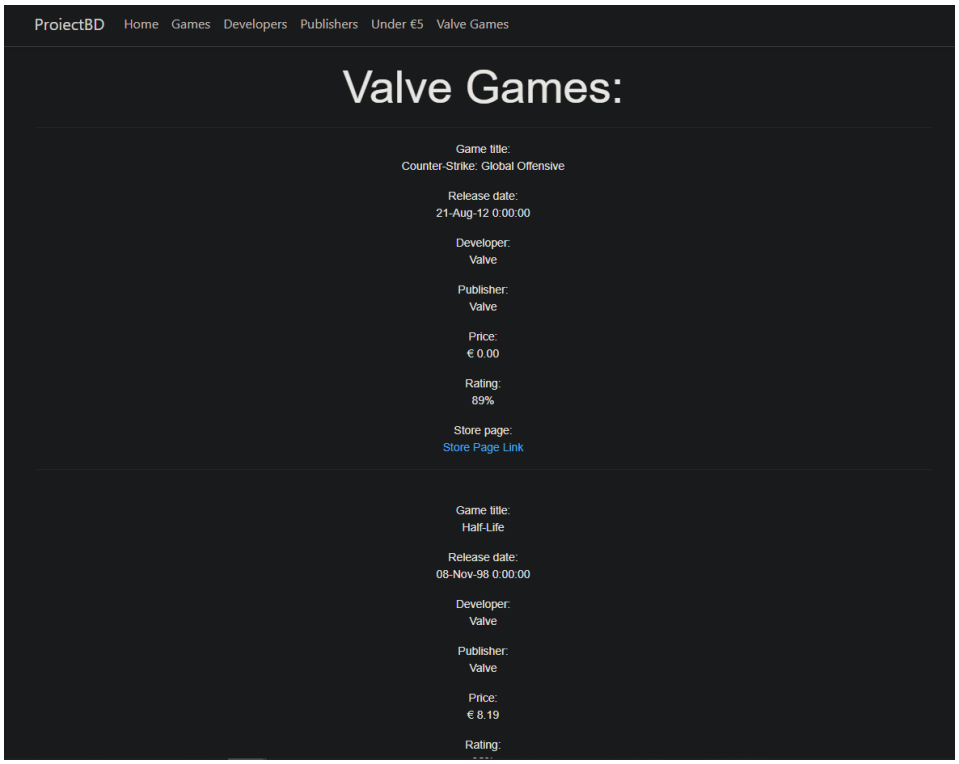


**Fig. 18 Raport 2**

- Concluzii
  Folosind baza de date relationala si framework-ul bazat pe C#/HTML, proiectul a fost unul relativ simplu de realizat, mai putin partea de aspect al frontend-ului, deoarece necesita cunostinte in HTML si foarte multă diligenţă. Cu toate acestea proiectul a ieşit până la capăt cu succes, si pot spune că am invatat multe lucruri noi.

- Resurse
  https://ocw.cs.pub.ro/courses/bd2
  https://alexpetrescu.net/
  https://docs.microsoft.com/en-us/sql/linux/quickstart-install-connect-docker?view=sql-server-ver15
  https://app.sqldbm.com/
  https://visualstudio.microsoft.com/vs/community/
  https://docs.microsoft.com/en-us/ef/core/cli/dotnet
  https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15
  https://store.steampowered.com/