Michael Laffan

Database Design Project

CMPT_308_902

November 22, 2021

Prof. Schwartz

Table of Contents

Video **games** today are a popular medium for people to enjoy and talk about with each other. A video game can be defined as an electronic game that involves human interaction with a user interface. A database can be used to look up different details about video games. For example, **developers** might want to know which **platforms** are the most popular to play video games on. In other circumstances, a gamer has a favorite developer and would want to know what other games they have developed, along with what platforms they are on. Also, a **publisher** would want to know what **cast members** have been cast in popular video games and would want to hire them. This database can be used to answer those questions and can also be used to look up things such as the names of platforms and what games have been rated as good by **reviewers**. The database can be created using these details:

-A video game, which can be uniquely identified by its id, has a genre, a price, a name, and an amount of copies that are sold. There are video games, however, that are never released, meaning that they are not rated by reviewers, have no cast members or publishers, and cannot be played on any platform. They did have one or more developers working on them, but they never completed them.

-Video games have developers that create and program them, who are uniquely identified by their id and have a name and a headquarters. Developers have parent companies that are publishers, who publish video games by marketing it to stakeholders and fund developers to create them. Not all publishers own developers though, since some video games have multiple publishers. Publishers are uniquely identified by their id and have a name and a location for their operations.

-Video games also have cast members who do voice overs for characters in them. However, there are some cast members who are not cast in video games, and some video games do not have cast members at all. Cast members have a name, a birth year, an age, and are uniquely identified by their id.

-Different platforms are created to allow these games to be played and are identified by their unique platform id. However, not every game can be played on every platform. These platforms also have a name, a price, and an amount of units sold.

-**Manufacturers** create and maintain these platforms, and they also have a unique id to be identified with. Manufacturers also have a name and a country that they are located in.

-Reviewer companies, who are uniquely identified by their id and have a name, rate video games with a numeric score.

The database must also be designed to fulfill the following requirements:

-A developer can create many video games, and a video game can have many developers.

-A publisher can own many developers. However, a developer can only be owned by one publisher.

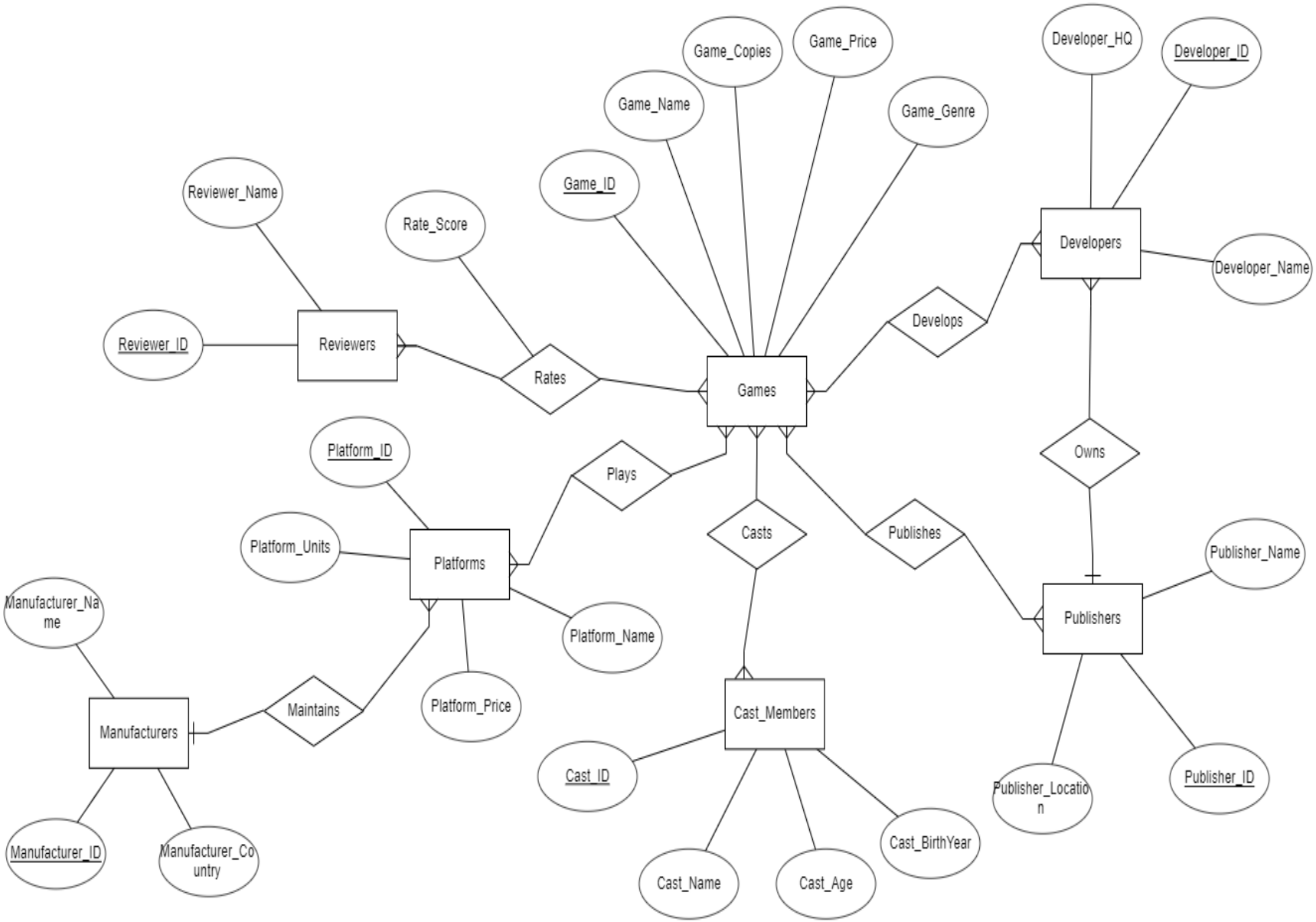
-A publisher can publish many games, and games can be published by many publishers.

-A cast member can do voice overs for characters in multiple games, and a game can cast multiple cast members.

-A game can be played on multiple platforms, and a platform can play many games.

-A manufacturer can maintain multiple platforms. However, a platform can only be maintained by one manufacturer.

-A game can be rated by many reviewers, and a reviewer can rate many video games.


Game_Copies
Game_Price
Developer_HQ
Developer_ID
Game_Name
Game_Genre
Reviewer_Name
Game_ID
Rate_Score
Developers
Developer_Name
Reviewer_ID
Reviewers
Develops
Rates
Games
Platform_ID
Plays
Owns
Platform_Units
Casts
Publishes
Publisher_Name
Platforms
Manufacturer_Name
Publishers
Platform_Name
Maintains
Platform_Price
Cast_Members
Manufacturers
Cast_ID
Publisher_Location
Publisher_ID
Manufacturer_ID
Manufacturer_Country
Cast_BirthYear
Cast_Name
Cast_Age

CREATE TABLE XGAMES

(

GAME_ID VARCHAR2(5),

GAME_NAME VARCHAR2(100),

GAME_COPIES NUMBER(12,0),

GAME_PRICE NUMBER(5,2),

GAME_GENRE CHAR(35),

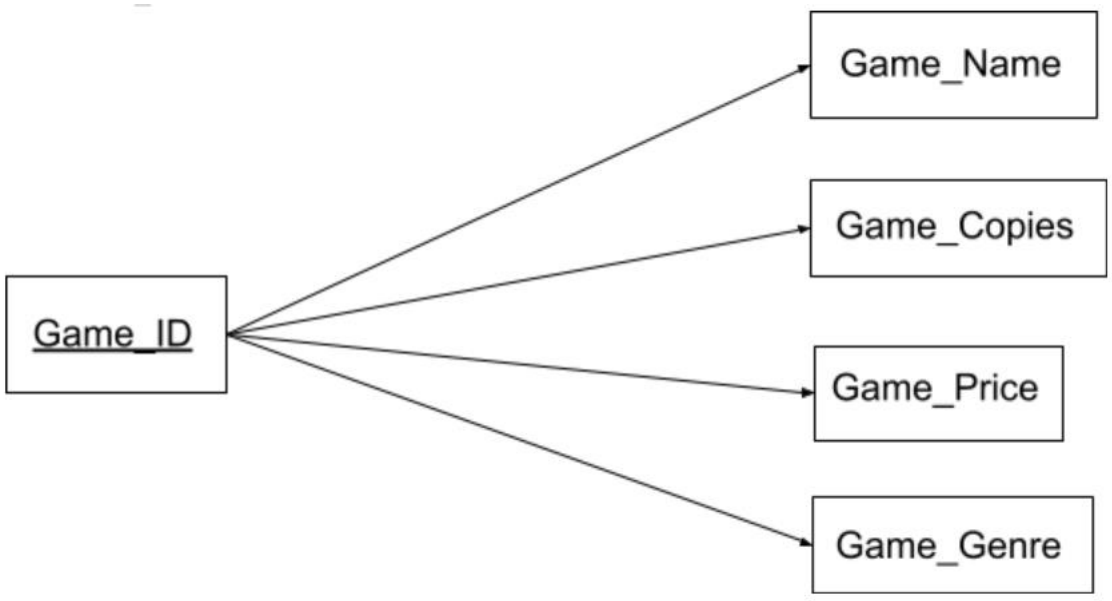
CONSTRAINT pk_XGAMES PRIMARY KEY (PUBLISHER_ID)

);

Description:

This table lists the ids, name, number of copies, price, and genre of certain video games, including ones that were cancelled.

3NF Justification:

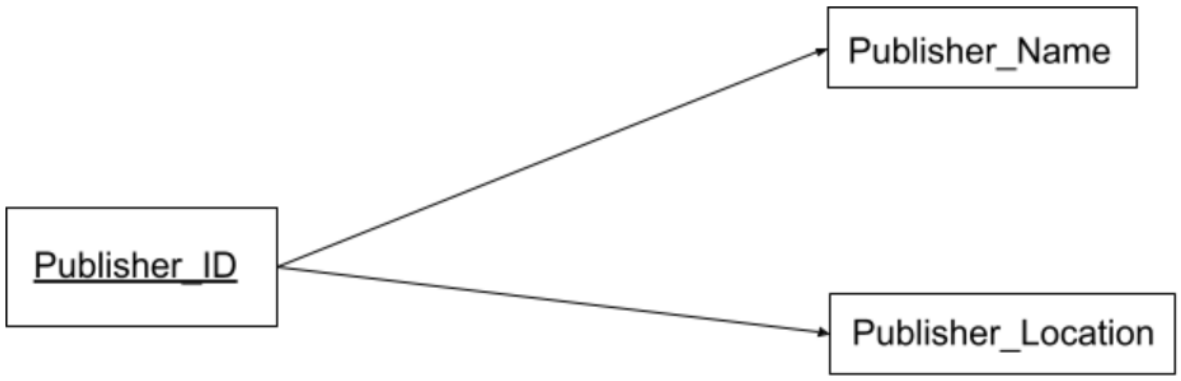


```
CREATE TABLE XPUBLISHERS
(
PUBLISHER_ID            VARCHAR2(5),
PUBLISHER_NAME          VARCHAR2(100),
PUBLISHER_LOCATION      CHAR(35),
CONSTRAINT pk_XPUBLISHERS PRIMARY KEY (PUBLISHER_ID)
);
```

Description:

This table lists the name, id, and locations of publishers that publish video games that are in the XGAMES table.

3NF Justification:



```
CREATE TABLE XPUBLISHES
(
PUBLISHER_ID        VARCHAR2(5),
GAME_ID             VARCHAR2(5),
CONSTRAINT pk_XPUBLISHES PRIMARY KEY (PUBLISHER_ID, GAME_ID),
CONSTRAINT fk_XPUBLISHES_XPUBLISHERS FOREIGN KEY (PUBLISHER_ID)
REFERENCES XPUBLISHERS(PUBLISHER_ID),
CONSTRAINT fk_XPUBLISHES_XGAMES FOREIGN KEY (GAME_ID) REFERENCES
XGAMES(GAME_ID)
);
```

Description:

This table connects the XGAMES and the XPUBLISHERS tables and lists the publisher ids along with the ids of the games they have published.

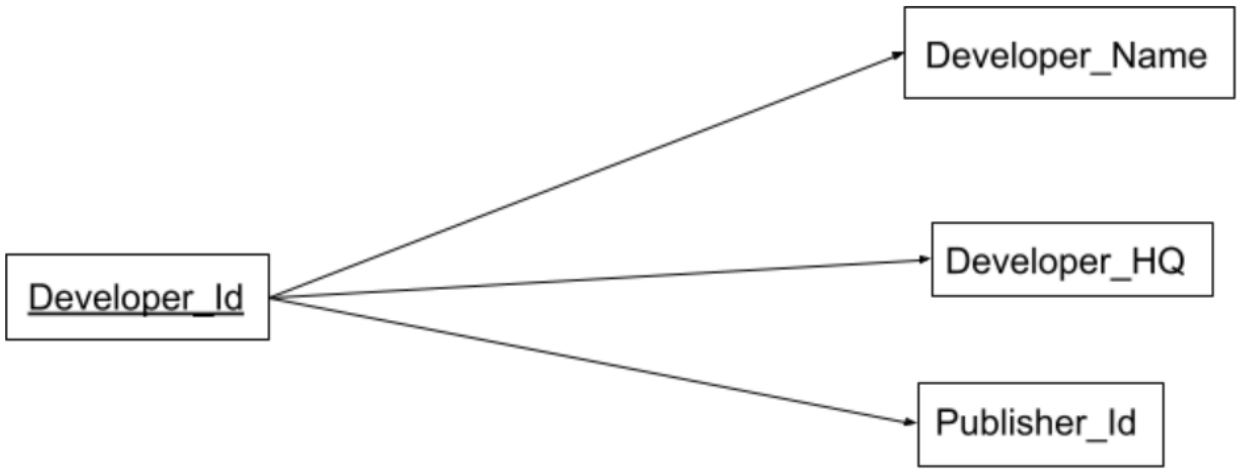
3NF Justification:

```
CREATE TABLE XDEVELOPERS
(
DEVELOPER_ID            VARCHAR2(5),
DEVELOPER_NAME          VARCHAR2(100),
DEVELOPER_HQ            CHAR(35),
PUBLISHER_ID            VARCHAR2(5),
CONSTRAINT pk_XDEVELOPERS PRIMARY KEY (DEVELOPER_ID),
CONSTRAINT fk_XDEVELOPERS FOREIGN KEY (PUBLISHER_ID) REFERENCES
XPUBLISHERS(PUBLISHER_ID)
);
```

Description:

This table lists the id, name, and headquarters of all the developers that have developed the video games that are listed in the XGAMES table. It also lists the ids of the publishers that own them.

3NF Justification:



```
CREATE TABLE XDEVELOPS
(
DEVELOPER_ID        VARCHAR2(5),
GAME_ID             VARCHAR2(5),
CONSTRAINT pk_XDEVELOPS PRIMARY KEY (DEVELOPER_ID, GAME_ID),
CONSTRAINT fk_XDEVELOPS_XDEVELOPERS FOREIGN KEY (DEVELOPER_ID)
REFERENCES XDEVELOPERS(DEVELOPER_ID),
CONSTRAINT fk_XDEVELOPS_XGAMES FOREIGN KEY (GAME_ID) REFERENCES
XGAMES(GAME_ID)
);
```

Description:

This table connects the XDEVELOPERS and XGAMES tables and lists the developer ids along with the ids of the games they have developed.

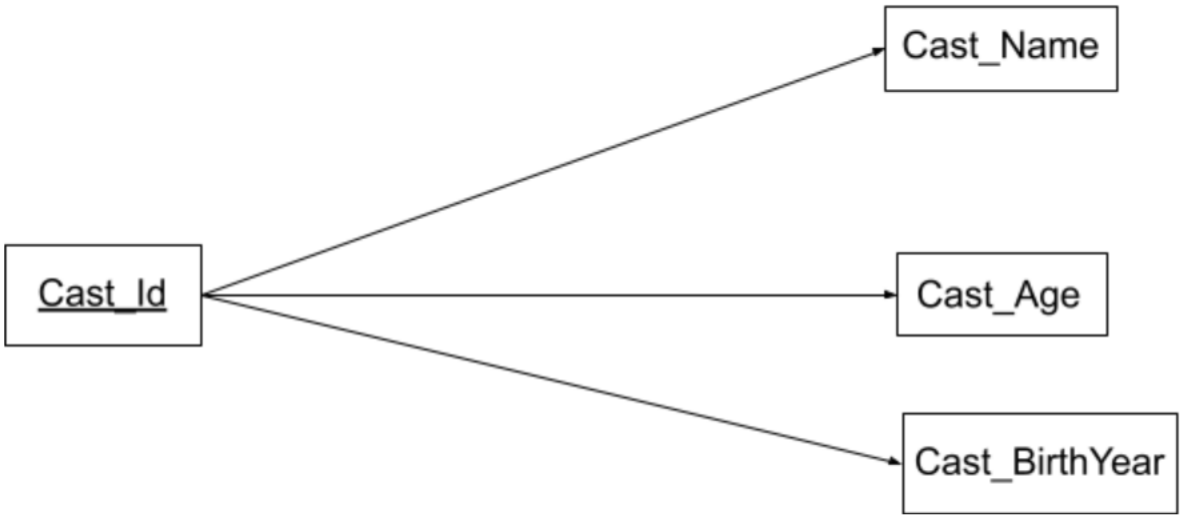
3NF Justification:

```
CREATE TABLE XCASTMEMBERS
(
CAST_ID              VARCHAR2(5),
CAST_NAME            CHAR(35),
CAST_AGE             NUMBER(2,0),
CAST_BIRTHYEAR CHAR(20),
CONSTRAINT pk_XCASTMEMBERS PRIMARY KEY (CAST_ID)
);
```

Description:

This table lists the id, name, age, and birth year of all cast members that have been cast in the video games listed in the XGAMES table as well as cast members who are not cast in the video games.

3NF Justification:

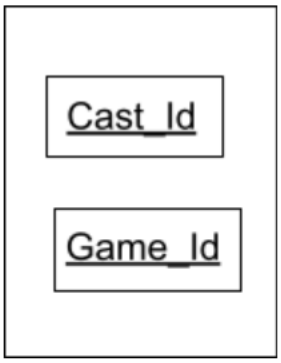


```
CREATE TABLE XCASTS
(
CAST_ID       VARCHAR2(5),
GAME_ID      VARCHAR2(5),
CONSTRAINT pk_XCASTS PRIMARY KEY (CAST_ID, GAME_ID),
CONSTRAINT fk_XCASTS_XCASTMEMBERS FOREIGN KEY (CAST_ID) REFERENCES
XCASTMEMBERS(CAST_ID),
CONSTRAINT fk_XCASTS_XGAMES FOREIGN KEY (GAME_ID) REFERENCES
XGAMES(GAME_ID)
);
```

Description:

This table connects the XCASTMEMBERS and the XGAMES tables and lists cast member ids along with the ids of the games they have been cast in.

3NF Justification:

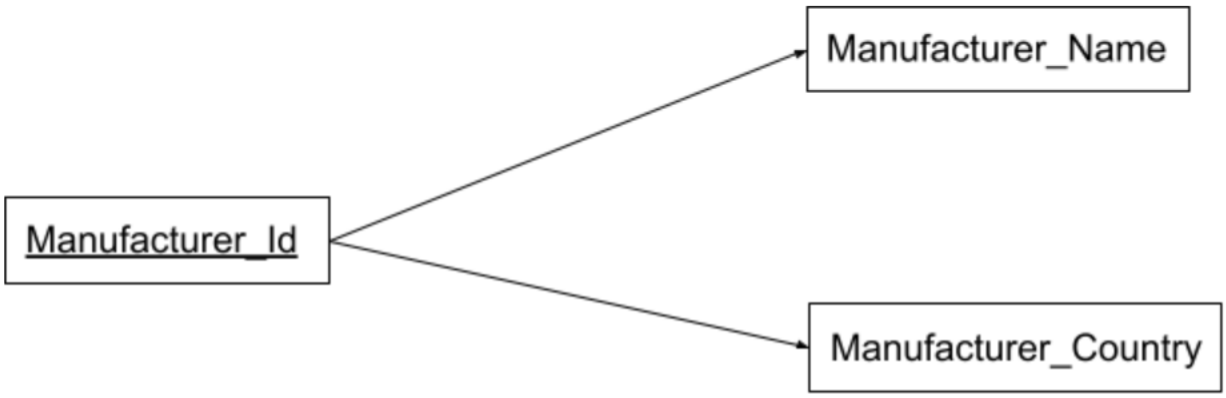


```
CREATE TABLE XMANUFACTURERS
(
MANUFACTURER_ID    VARCHAR2(5),
MANUFACTURER_NAME  CHAR(35),
MANUFACTURER_COUNTRY CHAR(35),
CONSTRAINT pk_XMANUFACTURERS PRIMARY KEY (MANUFACTURER_ID)
);
```

Description:

This table lists the id, name, and country location of all manufacturers that maintain a platform for video games to be played on.

3NF Justification:

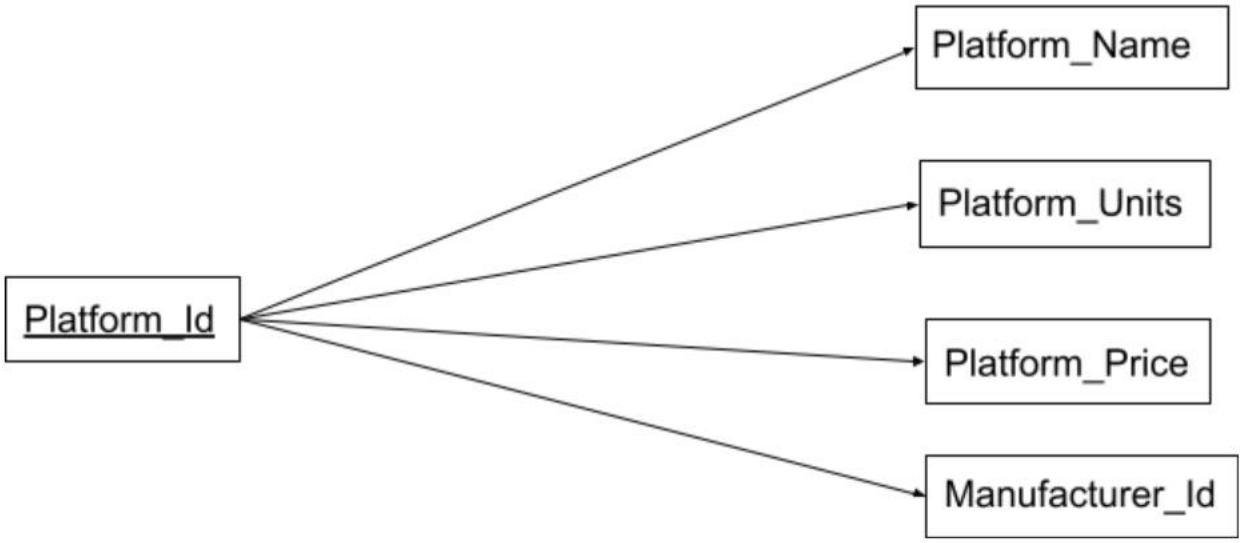


```
CREATE TABLE XPLATFORMS
(
PLATFORM_ID    VARCHAR2(5),
PLATFORM_NAME  VARCHAR2(35),
PLATFORM_UNITS NUMBER(13,0),
PLATFORM_PRICE NUMBER(5,2),
MANUFACTURER_ID VARCHAR2(5),
CONSTRAINT pk_XPLATFORMS PRIMARY KEY (PLATFORM_ID),
CONSTRAINT fk_XPLATFORMS_XMANUFACTURERS FOREIGN KEY
(MANUFACTURER_ID) REFERENCES XMANUFACTURERS(MANUFACTURER_ID)
);
```

Description:

This table lists the id, name, number of units, and price of all the platforms that allow video game to be played on. It also lists the id of the manufacturers that maintain them.

3NF Justification:



```
CREATE TABLE XPLAYS
(
PLATFORM_ID    VARCHAR2(5),
GAME_ID        VARCHAR2(5),
CONSTRAINT pk_XPLAYS PRIMARY KEY (PLATFORM_ID, GAME_ID),
CONSTRAINT fk_XPLAYS_XPLATFORMS FOREIGN KEY (PLATFORM_ID)
REFERENCES XPLATFORMS(PLATFORM_ID),
CONSTRAINT fk_XPLAYS_XGAMES FOREIGN KEY (GAME_ID) REFERENCES
XGAMES (GAME_ID)
);
```

Description:

This table connects the XPLATFORMS and the XGAMES tables and lists the ids of all the platforms as well as the ids of games that are played on them.

3NF Justification:

CREATE TABLE XREVIEWERS

(

REVIEWER_ID VARCHAR2(5),

REVIEWER_NAME CHAR(35),

CONSTRAINT pk_XREVIEWERS PRIMARY KEY (REVIEWER_ID)

);

Description:

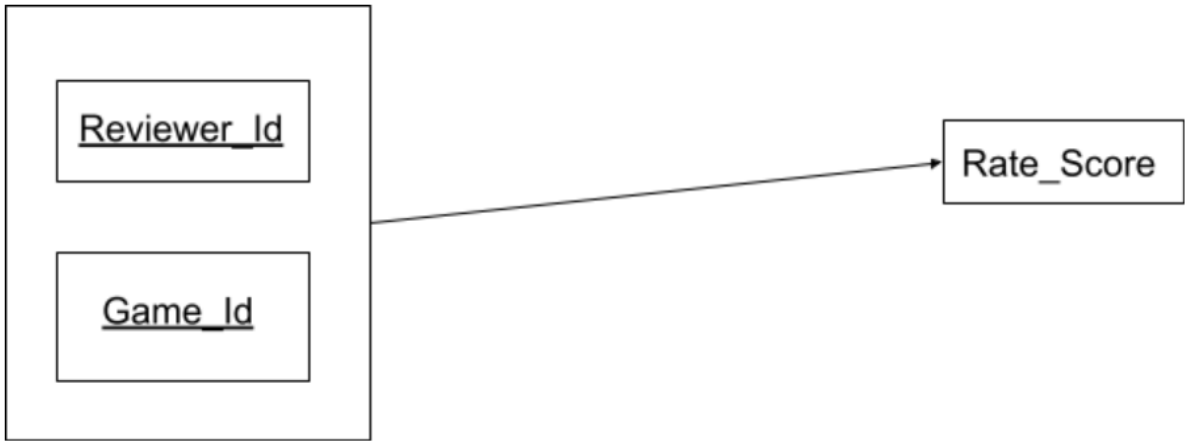
This table lists the ids and names of reviewer companies that rate video games with a numeric score.

3NF Justification:

```
CREATE TABLE XRATES
(
REVIEWER_ID VARCHAR2(5),
GAME_ID VARCHAR2(5),
RATE_SCORE NUMBER(3,1),
CONSTRAINT pk_XRATES PRIMARY KEY (REVIEWER_ID, GAME_ID),
CONSTRAINT fk_XRATES_XREVIEWERS FOREIGN KEY (REVIEWER_ID)
REFERENCES XREVIEWERS(REVIEWER_ID),
CONSTRAINT fk_XRATES_XGAMES FOREIGN KEY (GAME_ID) REFERENCES
XGAMES(GAME_ID)
);
```

This table connects the XREVIEWERS and XGAMES tables and lists the ids of the reviewers and the ids of the games they have rated, along with the score they have given the games as well.

3NF Justification:



1.Get the ids of games that can be played on every platform.

```
SELECT G.GAME_ID
FROM XGAMES G
WHERE NOT EXISTS
   (SELECT P.*
    FROM XPLATFORMS P
    WHERE NOT EXISTS
      (SELECT PL.*
       FROM XPLAYS PL
       WHERE G.GAME_ID = PL.GAME_ID
       AND PL.PLATFORM_ID = P.PLATFORM_ID));
```

	GAME_ID
1	G38
2	G41
3	G42
4	G72

Cardinality: 4

2.Get the ids of developers that only develop 'Action-Adventure' games.

```
SELECT D.DEVELOPER_ID
FROM XDEVELOPERS D
WHERE D.DEVELOPER_ID NOT IN
   (SELECT DV.DEVELOPER_ID
    FROM XDEVELOPS DV
    WHERE DV.GAME_ID NOT IN
      (SELECT G.GAME_ID
       FROM XGAMES G
       WHERE G.GAME_GENRE = 'Action-Adventure'));
```

	DEVELOPER_ID
1	D36
2	D61
3	D38
4	D41
5	D44
6	D11
7	D2
8	D33
9	D55
10	D17
11	D31
12	D22
13	D35
14	D20
15	D43
16	D7
17	D21
18	D26
19	D3
20	D47
21	D60

Cardinality: 21

3.Get the ids of cast members that are cast in none of the games with a price greater than 39.99.

```
SELECT C.CAST_ID
FROM XCASTMEMBERS C
WHERE C.CAST_ID IN
   (SELECT CA.CAST_ID
    FROM XCASTS CA
    WHERE CA.GAME_ID NOT IN
      (SELECT G.GAME_ID
       FROM XGAMES G
       WHERE G.GAME_PRICE > 39.99));
```

	CAST_ID
1	C24
2	C25
3	C66
4	C74
5	C16
6	C17
7	C18
8	C65
9	C26
10	C30
11	C6
12	C80
13	C27
14	C28
15	C8
16	C19
17	C29
18	C7
19	C36
20	C69
21	C70
22	C81
23	C75
24	C72
25	C73
26	C9

Cardinality: 26

4.Get the names and locations of all publishers, along with the names and headquarters of the developers they own, if any.

```
SELECT P.PUBLISHER_NAME, P.PUBLISHER_LOCATION, D.DEVELOPER_NAME,
D.DEVELOPER_HQ
FROM XPUBLISHERS P LEFT JOIN XDEVELOPERS D ON
P.PUBLISHER_ID = D.PUBLISHER_ID;
```

11	Capcom	Osaka	Capcom	Osaka
12	Capcom	Osaka	Highwire Games	Seattle
13	Capcom	Osaka	Atomic Games	Austin
14	Bethesda Softworks	Rockville	Tango Gameworks	Tokyo
15	Bethesda Softworks	Rockville	Bethesda Game Studios	Rockville
16	Bethesda Softworks	Rockville	Human Head Studios	Madison
17	Bethesda Softworks	Rockville	Panic Button Games	Austin
18	Bethesda Softworks	Rockville	idSoftware	Richardson
19	Bethesda Softworks	Rockville	Arkane Studios	Lyon
20	Focus Entertainment	Paris	Larian Studios	Oudenaarde
21	Larian Studios	Oudenaarde	(null)	(null)
22	Techland	Wroclaw	(null)	(null)
23	Warner Bros.	Burbank	Traveller's Tales	Knutsford
24	Warner Bros.	Burbank	Lucasarts	San Francisco
25	Warner Bros.	Burbank	Rocksteady Studios	London

28	Warner Bros.	Burbank	Techland	Wroclaw
29	Warner Bros.	Burbank	Platinum Games	Osaka
30	Ubisoft	Montreuil	Ubisoft Toronto	Toronto
31	Lionsgate Games	Santa Monica	Illumix	Redwood City
32	Lionsgate Games	Santa Monica	Phil Morg	Dallas
33	Lionsgate Games	Santa Monica	Steel Wool Studios	Oakland
34	Koei Tecmo	Kanagawa	Omega Force	Tochigi
35	ScottGames	Dallas	(null)	(null)
36	Rockstar Games	New York City	Rockstar North	Edinburgh
37	Koch Media	Hofen	Behaviour Interactive	Montreal
38	Deep Silver	Larkspur	4A Games	Sliema
39	Hello Games	Guildford	(null)	(null)
40	Paradox Interactive	Stockholm	Obsidian Entertainment	Irvine
41	Versus Evil	Baltimore	(null)	(null)
42	RockFish Games	Hamburg	RockFish Games	Hamburg

Cardinality: 68

5.Get the number of copies sold and genres of all video games, along with the rating scores they received and the ids of the reviewers that rated them, if any.

```
SELECT DISTINCT R.REVIEWER_ID, R.RATE_SCORE, G.GAME_COPIES,
G.GAME_GENRE
FROM XRATES R RIGHT JOIN XGAMES G ON
R.GAME_ID = G.GAME_ID;
```

35	R2	9.3	2000000	Action-Adventure
36	R1	9.1	2000000	Action-Adventure
37	R3	7	7000000	Action-Adventure
38	(null)	(null)	0	Action-Adventure
39	R3	8	1800000	First-Person Shooter

83	R2	9.5	12500000	Action-Adventure
84	R1	8.7	7000000	Action-Adventure
85	(null)	(null)	0	Horror
86	(null)	(null)	0	First-Person Shooter
87	R1	8.1	30710000	First-Person Shooter
88	R3	7	30710000	First-Person Shooter
89	R3	8	19000000	First-Person Shooter
90	R3	7	2000000	First-Person Shooter
91	R2	7	10000000	First-Person Shooter
92	R3	7	10000000	First-Person Shooter
93	R3	8	14300000	First-Person Shooter
94	R1	8.8	32494	Action-Adventure
95	R2	9	32494	Action-Adventure
96	R3	8	32494	Action-Adventure
97	R2	9	15000000	First-Person Shooter

139	R1	9.6	12500000	Action-Adventure
140	R3	9	12500000	Action-Adventure
141	R2	9.2	7000000	Action-Adventure
142	(null)	(null)	0	Simulator
143	R1	9.3	24770000	Fighting
144	R2	9.4	24770000	Fighting
145	R3	9	24770000	Fighting
146	R2	8	30710000	First-Person Shooter
147	R1	7.4	2000000	First-Person Shooter
148	R2	7.1	2000000	First-Person Shooter

Cardinality: 217

6.Get the names and ages of all cast members along with the names and prices of video games they were cast in, if any, and include names and prices of all video games along with the names and ages of the cast members they cast, if any.

```
SELECT C.CAST_NAME, C.CAST_AGE, G.GAME_NAME, G.GAME_PRICE
FROM XCASTMEMBERS C FULL JOIN XCASTS CA ON
C.CAST_ID = CA.CAST_ID FULL JOIN XGAMES G ON
CA.GAME_ID = G.GAME_ID;
```

44	Christine Lankin	42	Mass Effect: Andromeda	49.99
45	Alec Newman	46	Divinity: Original Sin	44.99
46	Anthony Gonzalez	17	Far Cry 6	59.99
47	Alex Fernandez	54	Far Cry 6	59.99
48	Christopher Judge	57	God of War	19.99
49	Tom Hanks	65	(null)	(null)
50	Leonardo DiCaprio	47	(null)	(null)
51	Will Smith	53	(null)	(null)
52	Denzel Washington	66	(null)	(null)
53	Johnny Depp	58	(null)	(null)

86	Kenji Nojima	45	Shadow of the Colossus	19.99
87	Naoki Bando	63	Shadow of the Colossus	19.99
88	Jeff Berg	74	Battlefield I	59.99
89	Mark Bonnar	52	Battlefield I	59.99
90	Thor Edgell	60	Battlefield 4	59.99
91	Kosha Engler	43	Battlefield 4	59.99
92	Anson Mount	48	The Evil Within	19.99
93	Jennifer Carpenter	41	The Evil Within	19.99
94	(null)	(null)	Scalebound	0
95	(null)	(null)	No Man's Sky	59.99
96	(null)	(null)	Super Mario Odyssey	39.99
97	(null)	(null)	Kirby's Return to D...	39.99
98	(null)	(null)	Arms	59.99
99	(null)	(null)	Prey 2	0
100	(null)	(null)	Origami King	59.99
101	(null)	(null)	Super Mario Maker 2	49.99
102	(null)	(null)	Mad Max	19.99
103	(null)	(null)	Disney Infinity 3.0	29.99
104	(null)	(null)	Lego Marvel's Avengers	29.99
105	(null)	(null)	Bayonetta	19.99

Cardinality: 142

7.Get the names of reviewers along with the rating scores of games that were given a score of less than 8 and created by a developer owned by a publisher located in Santa Monica.

```
SELECT DISTINCT R.REVIEWER_NAME, RA.RATE_SCORE
FROM XREVIEWERS R, XRATES RA, XGAMES G, XDEVELOPS DV, XDEVELOPERS D, XPUBLISHERS P
WHERE R.REVIEWER_ID = RA.REVIEWER_ID
AND RA.GAME_ID = G.GAME_ID
AND G.GAME_ID = DV.GAME_ID
AND DV.DEVELOPER_ID = D.DEVELOPER_ID
AND D.PUBLISHER_ID = P.PUBLISHER_ID
AND RA.RATE_SCORE < 8
AND P.PUBLISHER_LOCATION = 'Santa Monica';
```

	REVIEWER_NAME	RATE_SCORE
1	IGN	7
2	GameSpot	6.5
3	Metacritic	7.8
4	IGN	7.7
5	IGN	7.1
6	Metacritic	7.9
7	Metacritic	7.3
8	GameSpot	7
9	Metacritic	7.4

Cardinality: 9

8.Get the total number of games that can be played on consoles maintained by Nintendo.

```
SELECT COUNT(DISTINCT GAME_NAME) AS TOTAL_GAMES
FROM XGAMES G, XPLAYS P, XPLATFORMS PL, XMANUFACTURERS M
WHERE G.GAME_ID = P.GAME_ID
AND P.PLATFORM_ID = PL.PLATFORM_ID
AND PL.MANUFACTURER_ID = M.MANUFACTURER_ID
AND M.MANUFACTURER_NAME = 'Nintendo';
```

	TOTAL_GAMES
1	33

Cardinality: 1

9.Get the prices and the number of units sold for platforms that play games that have cast members whose age is greater than that of Troy Baker.

```
SELECT DISTINCT PLATFORM_PRICE, PLATFORM_UNITS
FROM XPLATFORMS PL, XPLAYS P, XGAMES G, XCASTS CA, XCASTMEMBERS C
WHERE PL.PLATFORM_ID = P.PLATFORM_ID
AND P.GAME_ID = G.GAME_ID
AND G.GAME_ID = CA.GAME_ID
AND CA.CAST_ID = C.CAST_ID
AND C.CAST_AGE >
(SELECT CM.CAST_AGE
 FROM XCASTMEMBERS CM
 WHERE CM.CAST_NAME = 'Troy Baker');
```

	PLATFORM_PRICE	PLATFORM_UNITS
1	250	101630000
2	300	92870000
3	140	400000000
4	500	51000000
5	400	116260000

Cardinality: 5

10.Get the ids, names, and countries of manufacturers whose platforms play games that were developed and published by companies in the same location.

```
SELECT DISTINCT M.MANUFACTURER_ID, M.MANUFACTURER_NAME,
M.MANUFACTURER_COUNTRY
FROM XMANUFACTURERS M, XPLATFORMS P, XPLAYS PL, XGAMES G,
XDEVELOPS DV, XDEVELOPERS D, XPUBLISHERS P, XPUBLISHES PB
WHERE M.MANUFACTURER_ID = P.MANUFACTURER_ID
AND P.PLATFORM_ID = PL.PLATFORM_ID
AND PL.GAME_ID = G.GAME_ID
AND G.GAME_ID = DV.GAME_ID
AND G.GAME_ID = PB.GAME_ID
AND DV.DEVELOPER_ID = D.DEVELOPER_ID
AND PB.PUBLISHER_ID = P.PUBLISHER_ID
AND D.DEVELOPER_HQ = P.PUBLISHER_LOCATION;
```

	MANUFACTURER_ID	MANUFACTURER_NAME	MANUFACTURER_COUNTRY
1	M3	Microsoft	United States
2	M1	Nintendo	Japan
3	M2	Sony	United States

Cardinality: 3