

Videogiochi per piattaforma

Alessio Pellegrino

0000873988

alessio.pellegrino@studio.unibo.it

William Arnone

0000874588

william.arnone@studio.unibo.it

Indice

| Argomento | Pagina |
|--|--------|
| 1. Analisi dei requisiti | 2 |
| a. Requisiti espressi in linguaggio naturale | 2 |
| b. <u>Glossario dei termini</u> | 3 |
| c. Eliminazione delle ambiguità presenti (sono state ridiscusse le specifiche) | 4 |
| d. Strutturazione dei requisiti | 4 |
| e. <u>Specifica Operazioni</u> | 6 |
| 2. Progettazione concettuale | 8 |
| a. Una prima Identificazione delle entità e relazioni (top-down) | 8 |
| b. Una migliore caratterizzazione delle entità di base (inside-out) | 8 |
| c. Estensione dello schema di base (bottom-up) | 9 |
| d. <u>Sviluppo componenti finali</u> | 10 |
| e. Unione nello schema completo | 12 |

| | |
|---|----|
| f. Dizionario dei Dati | 13 |
| g. <u>Regole aziendali</u> | 15 |
| 3. <u>Progettazione Logica</u> | 15 |
| a. <u>Tavole dei volumi e delle operazioni</u> | 15 |
| b. <u>Ristrutturazione dello schema concettuale</u> | 17 |
| c. Normalizzazione | 21 |
| d. <u>Traduzione verso il modello relazionale</u> | 22 |
| 4. Codifica SQL | 23 |
| a. Definizione dello schema | 23 |
| b. Creazione della vista | 25 |
| c. Codifica delle operazioni | 25 |
| 5. <u>Testing</u> | 31 |

1. Analisi dei requisiti

1a. Requisiti espressi in linguaggio naturale

Si vuole realizzare una base di dati per categorizzare i vari videogiochi e software sul mercato di cui si vogliono rappresentare i dati relativi al prodotto, le piattaforme su cui è presente, chi l'ha sviluppato e pubblicato.

In particolare di ogni prodotto, gioco o software che sia, si vogliono identificare il nome e la data di uscita (che assieme sono univoci), una descrizione sommaria dello stesso, una valutazione (da uno a cinque) ed il prezzo di listino.

Ogni compagnia si caratterizza per nome, data creazione e nazione, può poi possedere una piattaforma ed essere coinvolta in quanto publisher o sviluppatore in più prodotti. Per ogni piattaforma si vuole conoscere il nome, la data di uscita ed il supporto (se ancora è prodotta o giochi vengono sviluppati per essa), nel caso di console; mentre nel caso dello store PC si vuole identificare i sistemi operativi con cui è compatibile e il nome dello store. Ogni gioco può avere uno o più designer caratterizzati da nome, cognome, data di nascita, nazione ed identificativo. Inoltre, sia i

giochi che i software, possono avere uno o più categorie caratterizzate da un nome ed eventuali generi da cui derivano. Giochi e console possono infine raggruppati rispettivamente in saghe e generazioni: le prime comprendono la nascita della stessa, lo stato (in corso, finita), una descrizione ed il nome. Le generazioni, invece, hanno una data di uscita, stato (conclusa/ in corso), il numero identificativo e se è ancora disponibile all'acquisto.

1b. Glossario dei termini

| Termine | Descrizione | Sinonimi | Collegamenti |
|----------------|--|-----------------|-------------------------|
| Designer | Persona che si occupa della struttura di un videogioco | Autore | Gioco |
| Compagnia | Azienda che si occupa della creazione di un gioco o di una piattaforma | - | Prodotto, piattaforma |
| Sviluppatore | Compagnia che programma un software o un videogioco | - | Prodotto, Compagnia |
| Produttore | Compagnia che supervisiona e finanzia la produzione di un prodotto | - | Prodotto, Compagnia |
| Categoria | Genere o caratteristica di appartenenza di un videogioco | - | Categoria, Prodotto |
| Sottogenere | Categoria più dettagliata che specifica meglio le caratteristiche di un prodotto | - | Categoria |
| Saga | Insieme di videogiochi dello stesso universo narrativo | Franchise | Gioco, Raggruppamento |
| Generazione | Insieme di console uscite nello stesso periodo, con simili caratteristiche tecniche e giochi in comune | - | Console, Raggruppamento |
| Raggruppamento | Insieme di giochi o console con caratteristiche comuni | - | Generazione, saga |
| Piattaforma | Hardware o programma che è necessario per eseguire un | - | Compagnia |

| | | | |
|------------|---|-----------------------|--|
| | gioco o un programma | | |
| Store | Negoziato virtuale che mette a disposizione giochi e programmi. Necessario per la loro esecuzione | Negoziato, launcher | Prodotto |
| Console | Hardware che permette di eseguire videogiochi creati apposta per essa | - | Generazione |
| Gioco | Programma a scopo ludico | Videogioco, videogame | Designer, saga, console, prodotto |
| Software | Programma acquistabile su uno store che non sia un videogioco | - | Prodotto |
| Prodotto | Gioco o software messo in vendita su uno store o per una console | Programma | Categoria, recensione, sviluppatore, produttore, store |
| Recensione | Voto per un prodotto | Giudizio | Prodotto, utente |
| Utente | Persona fisica che usa un prodotto | - | Recensione |

1c. Eliminazione delle ambiguità presenti (sono state ridiscusse le specifiche)

La recensione non è un semplice attributo del prodotto ma viene assegnata da un utente che ne scrive una personale ed è quindi da attribuire a chi la scrive.

Normalmente i videogiochi sono considerati software ma, in questo caso essendo loro una parte importante del progetto, è stato deciso di considerare software tutto ciò che non è un videogioco

I software non sono disponibili su console

L'unica generazione attiva è l'ultima presente.

I requisiti di un prodotto possono variare a seconda dello store di riferimento, quindi si riferiscono ai requisiti indicati da uno store per un determinato prodotto.

1d. Strutturazione dei requisiti

- Frasi di carattere generale

Si vuole realizzare una base di dati per categorizzare i vari videogiochi e software sul mercato di cui si vogliono rappresentare i dati relativi al prodotto, le piattaforme su cui è presente, chi l'ha sviluppato e pubblicato

- Frasi relative ai software ed ai videogiochi

In particolare di ogni prodotto, gioco o software che sia, si vogliono identificare il nome e la data di uscita (che assieme sono univoci), una descrizione sommaria dello stesso, una valutazione (da uno a cinque) ed il prezzo di listino. Inoltre, sia i giochi che i software, possono avere uno o più categorie caratterizzate da un nome ed eventuali generi da cui derivano

- Frasi relative alle compagnie

Ogni compagnia si caratterizza per nome, data creazione e nazione, può poi possedere una piattaforma ed essere coinvolta in quanto publisher o sviluppatore in più prodotti

- Frasi relative alle piattaforme

Per ogni piattaforma si vuole conoscere il nome, la data di uscita ed il supporto (se ancora è prodotta o giochi vengono sviluppati per essa), nel caso di console; mentre nel caso dello store PC si vuole identificare i sistemi operativi con cui è compatibile e il nome dello store.

- Frasi relative ai designer

Ogni gioco può avere uno o più designer caratterizzati da nome, cognome, data di nascita, nazione ed identificativo

- Frasi relative ai raggruppamenti

Giochi e console possono infine raggruppati rispettivamente in saghe e generazioni: le prime comprendono la nascita della stessa, lo stato (in corso, finita), una descrizione ed il nome. Le generazioni, invece, hanno una data di uscita, stato (conclusa/ in corso), il numero identificativo e se è ancora disponibile all'acquisto

- Frasi relative agli user ed alle recensioni

Per poter dare un voto al gioco o software ogni utente deve avere un nickname unico ed esprimere il proprio voto esclusivamente per un gioco per volta

1e. Specifica Operazioni

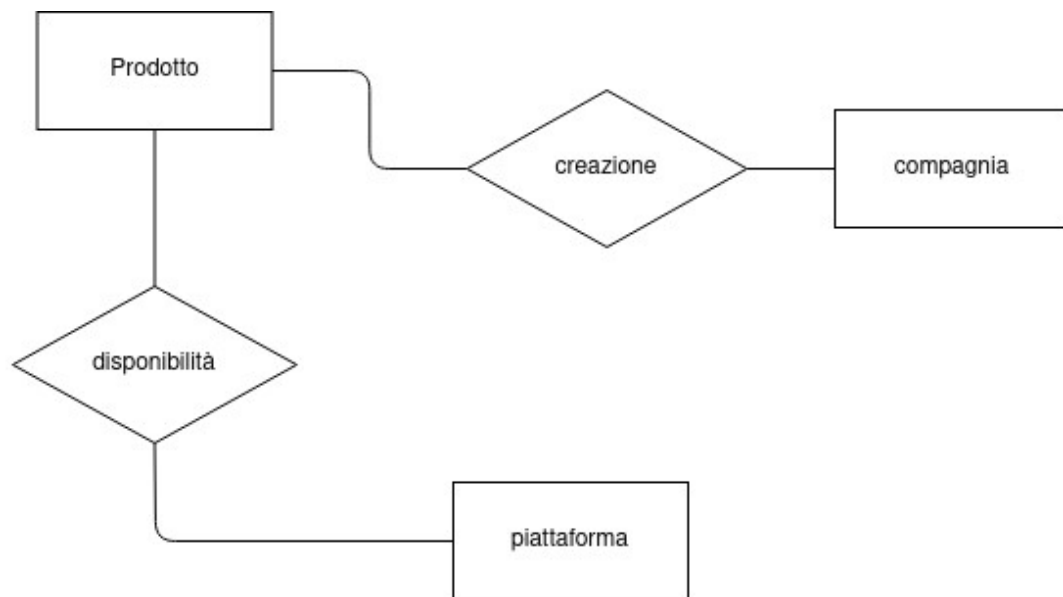
1. inserire una categoria (1 volta l'anno)
2. inserire una compagnia (1 volta l'anno)
3. inserire una console (1 volta ogni 4 anni)
4. inserire un designer (1 volta l'anno)
5. inserire la disponibilità di un prodotto per uno store PC (5 volte al mese)
6. inserire la riproducibilità di un gioco per una console (5 volte al mese)
7. inserire un software e la sua categoria d'appartenenza (3 volte al mese)
8. inserire un gioco con designer che ci hanno lavorato, categorie e saghe di appartenenza (5 volte al mese)
9. inserire una recensione (5 volta al giorno)
10. inserire una saga (1 volta l'anno)
11. inserire uno store (1 volta ogni 3 anni)
12. inserire un utente (1 volta al giorno)
13. visualizzare tutte le categorie (100 volte al giorno)
14. visualizzare i prodotti di una categoria (200 volte al giorno)
15. visualizzare tutte le compagnie (50 volte al giorno)
16. visualizzare tutte le console (100 volte al giorno)
17. visualizzare tutti i designer (50 volte al giorno)
18. visualizzare giochi di una console ordinati per valutazione (200 volte al giorno)
19. visualizzare giochi di uno store (300 volte al giorno)
20. visualizzare tutti i prodotti (500 volte al giorno)
21. visualizzare tutti i giochi ordinati per valutazione (500 volte al giorno)

22. visualizzare tutti i software (200 volte al giorno)
23. visualizzare i designer che hanno lavorato ad un gioco (50 volte al giorno)
24. visualizzare tutte le saghe ordinate per anno di nascita(100 volte al giorno)
25. visualizzare tutti i giochi di una saga (50 volte al giorno)
26. visualizzare tutti i produttori (50 volte al giorno)
27. visualizzare tutti gli sviluppatori (50 volte al giorno)
28. visualizzare tutti gli store di una compagnia (10 volte al giorno)
29. visualizzare tutte le console di una compagnia (10 volte al giorno)
30. visualizzare tutte le recensioni di un utente (10 volte al giorno)
31. aggiornare un voto di una recensione (1 volta l'anno)
32. aggiornare il prezzo di un gioco (1 volta al mese)
33. aggiornare la saga di un gioco (2 volta l'anno)
34. visualizzare tutti i sottogeneri di una categoria (50 volte al giorno)
35. visualizzare le compagnie con cui ha lavorato un designer (10 volte al giorno)
36. Visualizzare tutte le generazioni supportate con la loro data di nascita (10 volte al giorno)
37. visualizzare i software di un'azienda ordinati per prezzo (50 volte al giorno)
38. visualizzare chi ha prodotto i giochi sviluppati da una compagnia (10 volte al giorno)
39. aggiungere una nuova tipologia ad un prodotto (5 volte al mese)
40. aggiornare la lista di designer che hanno lavorato ad un gioco (3 volte al mese)
41. aggiornare il supporto di una determinata console (2 volta ogni 7 anni)

NOTA: ogni visualizzazione di prodotti è accompagnata dal suo voto calcolato come media dei voti di tutte le recensioni

2. Progettazione concettuale

2a. Una prima Identificazione delle entità e relazioni (top down)



Seguendo la specifica sono state identificate le seguenti entità e relazioni di base attorno alle quali costruire il resto della rappresentazione.

dove, per ogni prodotto, si specificano le piattaforme in cui è reperibile e quale sia la compagnia che lo produce.

2b. Una migliore caratterizzazione delle entità di base (inside-out)

L'entità Prodotto viene ora espansa in 2 sotto entità: Gioco e Software che meglio rappresentano le specifiche.

Anche Piattaforma viene dettagliata in Store Pc e Console per le stesse motivazioni.

Vengono anche meglio caratterizzate le relazioni, dividendo la relazione “creazione” tra prodotto e “Compagnia” in “Sviluppatore” e “Produttore” e aggiungendo la relazione “riproducibilità” tra “Gioco” e “Console” per rappresentare la possibilità che un gioco vada su una console (cosa

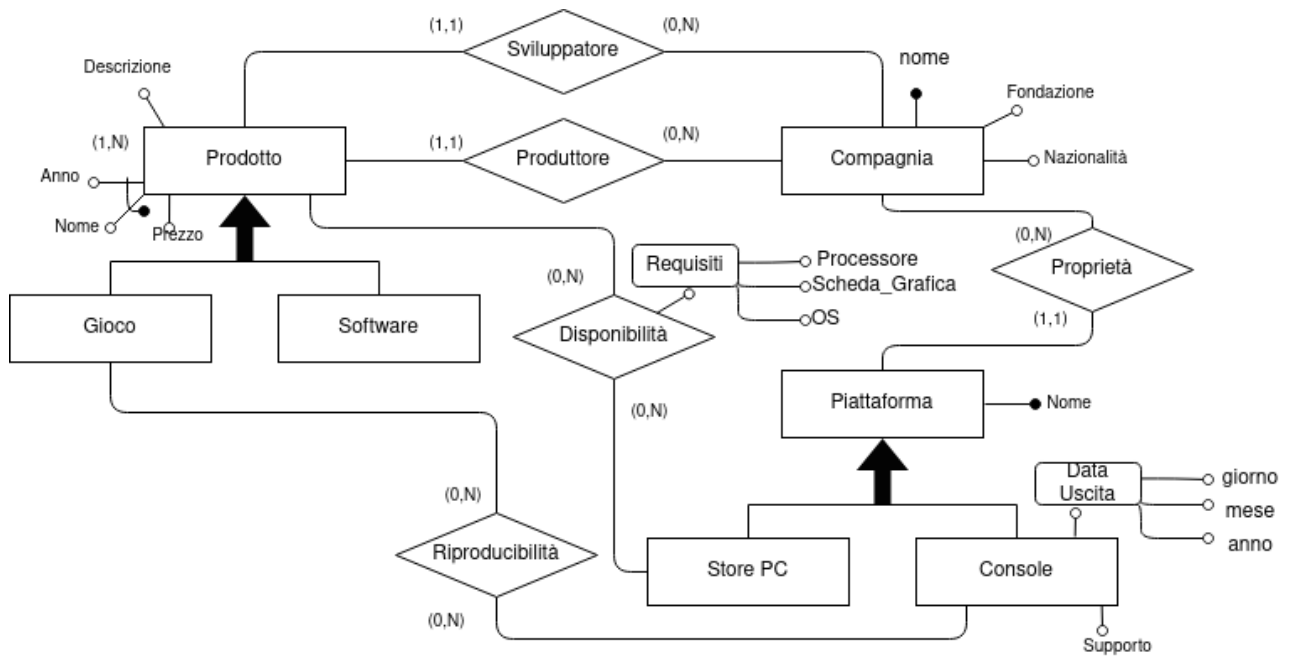
esclusiva dei giochi e non dei software) e la relazione “Proprietà” tra ”Compagnia” e “Piattaforma” per rappresentare la possibilità che una compagnia posseda una piattaforma.

Come attributi sono stati identificati nome e anno (insieme chiave), prezzo e descrizione per le entità Gioco e Software (attributi comuni ad entrambe).

Compagnia ha l’attributo Nome come chiave e fondazione e nazionalità come attributi aggiuntivi.

La piattaforma ha l’attributo nome come chiave, inoltre la console ha due attributi aggiuntivi: data uscita e supporto.

In fine la relazione disponibilità ha un attributo aggiuntivo “requisiti” per meglio dettagliare i requisiti di sistema minimi necessari per eseguire il prodotto (indicati dallo store).



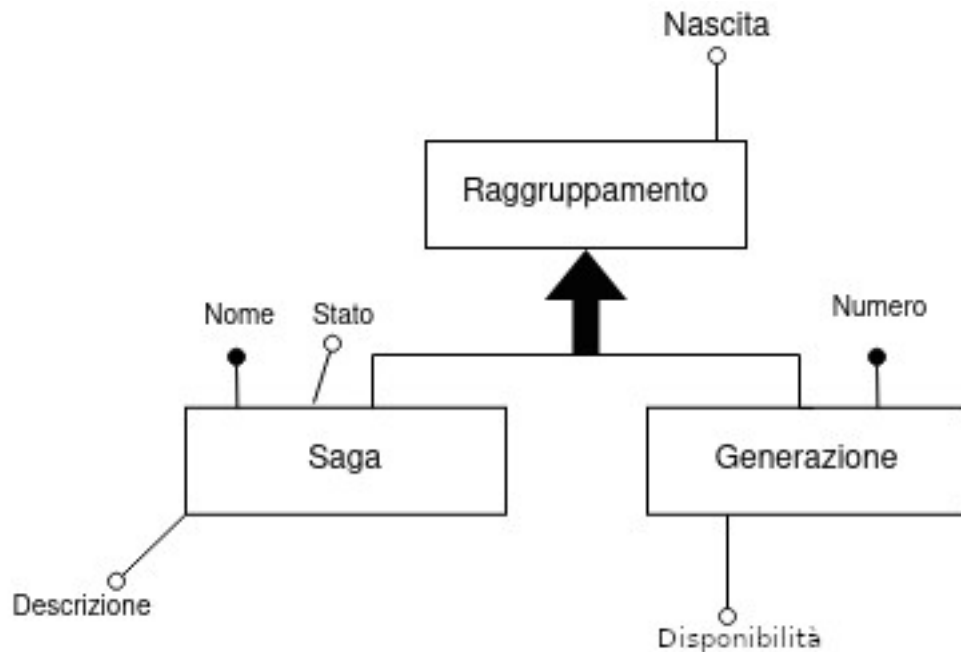
2c. Estensione dello schema di base (bottom-up)

Dopo avere definito le precedenti entità e relazioni di base abbiamo utilizzato un approccio bottom-up per catturare le caratteristiche mancanti.

Rifacendoci alle specifiche sono state identificate le seguenti entità: Categoria, Designer, Utente e Raggruppamento.

2d. Sviluppo componenti finali (inside-out)

Raggruppamento:

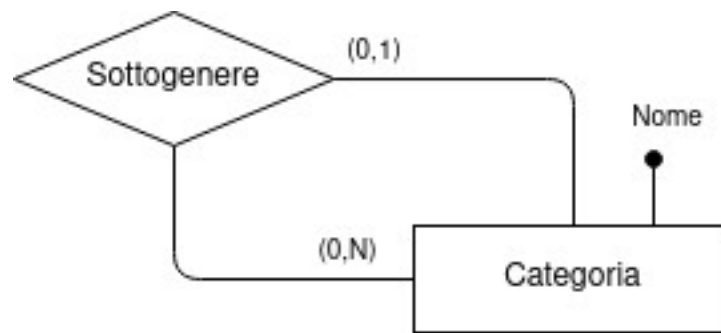


Analizzando l'entità Raggruppamento si è deciso di derivarne due entità: Saga e Generazione. Entrambe le entità condividono l'attributo nascita. (per indicare quando raggruppamento è nato)

Saga, esclusiva per i videogiochi, ha come attributi propri Nome (univoco), stato che indica se sono previsti ulteriori giochi ed è quindi una saga attiva oppure è terminata e Descrizione; raccoglie tutti i videogiochi appartenenti allo stesso franchise.

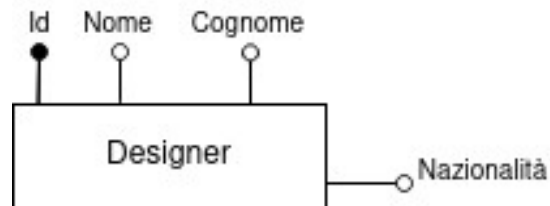
Generazione, esclusiva per le console, ha gli attributi: Numero (identificate da un'enumerazione progressiva) e Disponibilità; raccoglie le console uscite in un certo lasso di tempo e con caratteristiche simili.

Categoria:



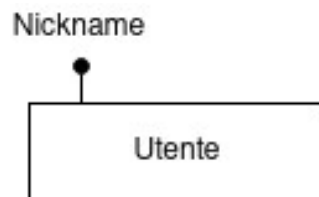
La categoria è caratterizzata da un nome univoco e rappresenta il genere o una caratteristica del prodotto, la categoria può avere o meno più sottogeneri, quindi una categoria può derivare da un'altra.

Designer:



Un designer ha Nome, Cognome e Nazionalità, non essendo univoca nessuna combinazione di queste tre si è deciso di attribuire un identificativo numerico.

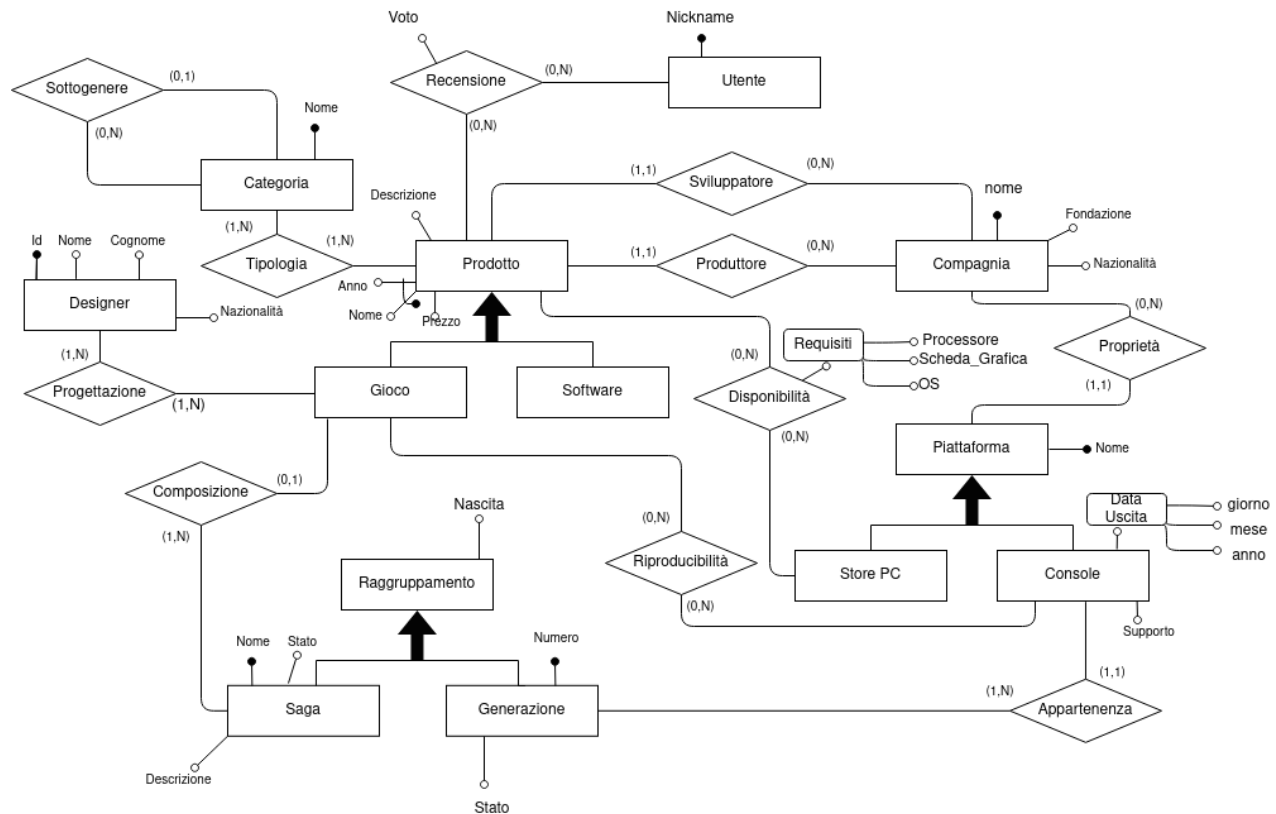
Utente:



L'Utente possiede solo un Nickname, il quale deve essere univoco.

2e. Unione nello schema completo

Le ultime entità sono state collegate tramite le relazioni: Composizione (Saga-Gioco), Recensione (Utente-Prodotto), Progettazione (Designer-Gioco), Tipologia (Categoria-Prodotto) e Appartenenza (Generazione-Console).



2f. Dizionario dei Dati

Entità:

| Nome | Descrizione | Attributi | Identificatore |
|----------------|---|---|----------------------------------|
| Categoria | Genere o caratteristica di un prodotto | - | Nome(stringa) |
| Utente | Persona che valuta un prodotto | - | Nickname(stringa) |
| Prodotto | Programma che può essere acquistato e valutato | Prezzo(numerico), Descrizione(stringa) | Nome(stringa), Anno(numerico) |
| Gioco | Tipologia di prodotto con scopo ludico | - | - |
| Software | Prodotto dai diversi scopi meno quelli ludici | - | - |
| Compagnia | Azienda lavora nell'ambito informatico | Fondazione(numerico), Nazionalità(stringa) | Nome(stringa) |
| Designer | Professionista nell'ambito della progettazione di videogiochi | Nome(stringa), Cognome(stringa), Nazionalità(stringa) | Id(numerico) |
| Piattaforma | Software/Hardware che permette di eseguire un prodotto | - | Nome(stringa) |
| Store | Negoziario virtuale necessario all'esecuzione di un prodotto | - | - |
| Console | Hardware sulla quale è eseguito un videogioco | Data Uscita(data), Supporto(booleano) | - |
| Raggruppamento | Insieme di elementi dalle caratteristiche comuni | Stato: Terminata, in corso; Nascita(numerico) | - |
| Saga | Insieme di videogiochi appartenenti allo stesso franchise | Descrizione(stringa) | Nome(stringa) |
| Generazione | Insieme di console con caratteristiche simili prodotte nello stesso periodo | Disponibilità(booleano) | Numero(numerico) |

Relazioni:

| Nome | Descrizione | Entità Coinvolte | Attributi |
|-----------------|---|------------------------------------|--------------------|
| Sottogenere | Associa una categoria ad un'altra categoria dalla quale deriva | Categoria(0,1) Categoria(0,N) | - |
| Recensione | Associa un utente ad un prodotto assegnandogli un voto | Utente(0,N) Prodotto(0,N) | Voto(numerico) |
| Tipologia | Associa i prodotti con le categorie che possiede | Categoria(1,N) Prodotto(1,N) | - |
| Sviluppatore | Associa un prodotto con la compagnia che lo ha sviluppato | Prodotto(1,1) Compagnia(0,N) | - |
| Produttore | Associa un prodotto con la compagnia che si è occupata della produzione | Prodotto(1,1) Compagnia(0,N) | - |
| Progettazione | Associa un Gioco con i Designer che ci hanno lavorato | Designer(1,N) Gioco(1,N) | - |
| Disponibilità | Associa un Prodotto con gli Store su cui è disponibile | Prodotto(0,N) Store(1,N) | Requisiti(stringa) |
| Proprietà | Associa una Piattaforma alla compagnia che ne detiene i diritti o che l'ha creata | Compagnia(0,N) Piattaforma(1,1) | - |
| Composizione | Associa una Saga con i Giochi che la compongono | Gioco(0,1) Saga(1,N) | - |
| Riproducibilità | Associa un Gioco con le Console su cui può essere riprodotto | Gioco(0,N) Console(1,N) | - |
| Appartenenza | Associa una Console con la Generazione alla quale appartiene | Generazione(1,N) Console(1,1) | - |

2g. Regole aziendali

Regole di vincolo

- ◆ Il voto è compreso tra 1 e 5
- ◆ Il prezzo di un prodotto è positivo, a due cifre decimali ed espresso in euro
- ◆ Supporto è vero se la Console è supportata

Regole di derivazione

- Le aziende per cui ha lavorato un Designer si ottengono combinando i Giochi che ha progettato con le compagnie che li hanno sviluppati
- La valutazione di un gioco si ottiene tramite la media delle recensioni
- L'attributo Disponibilità è vero se ci sono Console supportate di quella Generazione

3. Progettazione Logica

3a. Tavole dei volumi e delle operazioni

Tavola dei volumi:

| Concetto | Tipo | Volume |
|----------------|------|--------|
| Categoria | E | 250 |
| Utente | E | 1000 |
| Prodotto | E | 6000 |
| Gioco | E | 4000 |
| Software | E | 2000 |
| Designer | E | 200 |
| Compagnia | E | 60 |
| Piattaforma | E | 50 |
| Store | E | 20 |
| Console | E | 30 |
| Raggruppamento | E | 38 |
| Saga | E | 30 |
| Generazione | E | 8 |
| Sottogenere | R | 100 |
| Tipologia | R | 12000 |
| Recensione | R | 8000 |

| | | |
|-----------------|---|------|
| Sviluppatore | R | 6000 |
| Produttore | R | 6000 |
| Proprietà | R | 50 |
| Disponibilità | R | 4000 |
| Progettazione | R | 8000 |
| Composizione | R | 1000 |
| Riproducibilità | R | 3000 |
| Appartenenza | R | 30 |

Tavola delle operazioni:

| | |
|----|---------------------|
| 1 | 1 volta l'anno |
| 2 | 1 volta l'anno |
| 3 | 1 volta ogni 4 anni |
| 4 | 1 volta l'anno |
| 5 | 5 volte al mese |
| 6 | 5 volte al mese |
| 7 | 3 volte al mese |
| 8 | 5 volte al mese |
| 9 | 5 volte al giorno |
| 10 | 1 volta l'anno |
| 11 | 1 volta ogni 3 anni |
| 12 | 1 volta al giorno |
| 13 | 100 volte al giorno |
| 14 | 200 volte al giorno |
| 15 | 50 volte al giorno |
| 16 | 100 volte al giorno |
| 17 | 50 volte al giorno |
| 18 | 200 volte al giorno |
| 19 | 300 volte al giorno |
| 20 | 500 volte al giorno |
| 21 | 500 volte al giorno |
| 22 | 200 volte al giorno |
| 23 | 50 volte al giorno |

| | |
|----|---------------------|
| 24 | 100 volte al giorno |
| 25 | 50 volte al giorno |
| 26 | 50 volte al giorno |
| 27 | 50 volte al giorno |
| 28 | 10 volte al giorno |
| 29 | 10 volte al giorno |
| 30 | 10 volte al giorno |
| 31 | 1 volta l'anno |
| 32 | 1 volta al mese |
| 33 | 2 volte l'anno |
| 34 | 50 volte al giorno |
| 35 | 10 volte al giorno |
| 36 | 10 volte al giorno |
| 37 | 50 volte al giorno |
| 38 | 10 volte al giorno |
| 39 | 5 volte al mese |
| 40 | 3 volte al mese |
| 41 | 2 volta ogni 7 anni |

3b. Ristrutturazione dello schema concettuale

Eliminazione delle ridondanze

L'attributo "Disponibilità" nell'entità "Generazione" indica se esiste ancora una console supportata di quella generazione ma la stessa informazione si può ottenere controllando l'attributo "Supporto" di ogni console appartenente alla generazione.

L'attributo "nascita" nei raggruppamenti indica quando è uscito il primo di una saga o quando è uscita la prima console di una generazione; ma la stessa informazione si può ottenere controllando il più vecchio gioco (o console) appartenente al raggruppamento scelto.

Tavola degli accessi in presenza di ridondanze:

| Operazione 24 | | | |
|----------------------|------------------|----------------|-------------|
| Concetto | Costrutto | Accessi | Tipo |
| Saga | Entità | 30 | L |

| Operazione 36 | | | |
|----------------------|------------------|----------------|-------------|
| Concetto | Costrutto | Accessi | Tipo |
| Generazione | Entità | 8 | L |

Tavola degli accessi in assenza di ridondanze:

| Operazione 24 | | | |
|----------------------|------------------|----------------|-------------|
| Concetto | Costrutto | Accessi | Tipo |
| Saga | Entità | 30 | L |
| Gioco | Entità | 1000 | L |
| Composizione | Relazione | 1000 | L |

| Operazione 36 | | | |
|----------------------|------------------|----------------|-------------|
| Concetto | Costrutto | Accessi | Tipo |
| Generazione | Entità | 8 | L |
| Console | Entità | 30 | L |
| Appartenenza | Relazione | 30 | L |

In presenza di ridondanza il costo delle varie operazioni (considerando il doppio costo di una scrittura rispetto a una lettura):

$$\text{Op.24} = 30(\text{costo}) * 100(\text{volte al giorno}) = 3000$$

$$\text{Op.36} = 8(\text{costo}) * 10(\text{volte al giorno}) = 80$$

$$\text{Totale} = 3080$$

In assenza di ridondanza il costo delle varie operazioni (considerando il doppio costo di una scrittura rispetto a una lettura):

Op.24 = 2030(costo) *100 (volte al giorno) = 203000

Op.36 = 68(costo) *10 (volte al giorno) = 680

Totale = 203680

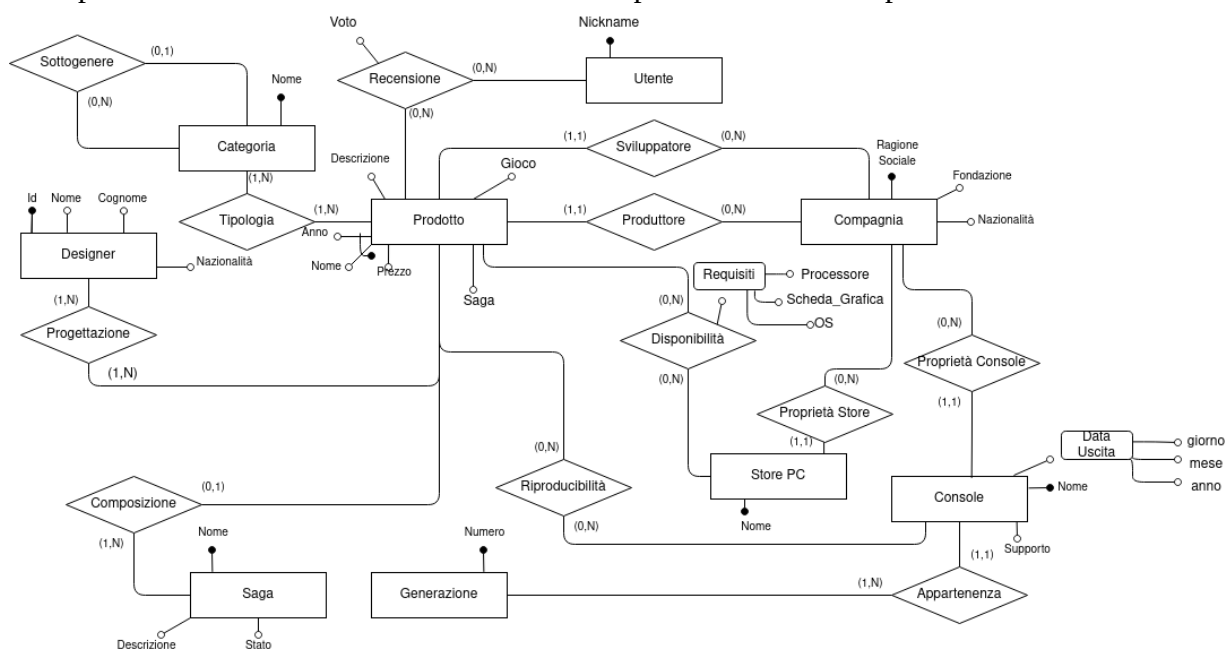
Eliminazione delle gerarchie:

Data la simile natura degli accessi tra “Prodotto”, “Gioco” e “Software”, si è deciso di unire le tre entità, aggiungendo un campo “gioco” booleano per distinguere tra gioco e software.

(RV6) solo i giochi possono avere saghe, quindi solo chi ha l’attributo gioco a True può avere la relazione

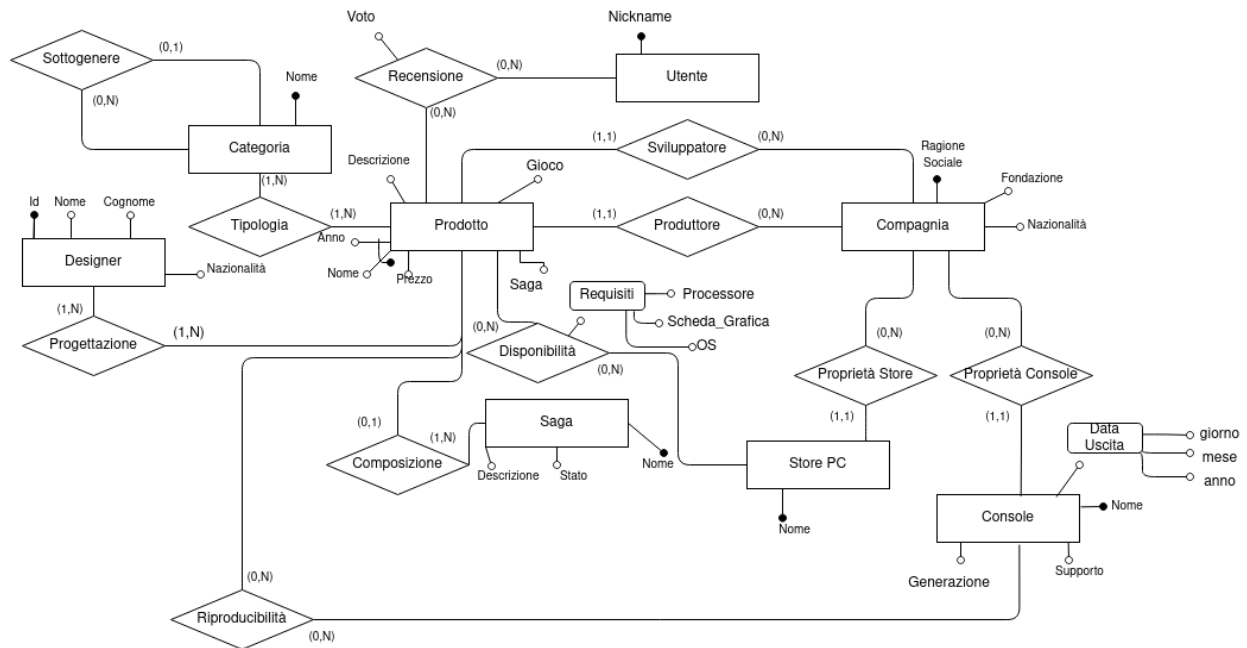
Riguardo Saga e Generazione, si è preferito distinguerle ulteriormente, aggiungendo gli attributi “Stato” e “Nascita” ad entrambe ed eliminando l’entità “Raggruppamento”. Questo principalmente perché sono accedute in maniera profondamente diversa.

Lo stesso approccio è stato utilizzato anche per “Piattaforma”. “Store” e “Console” vengono per lo più accedute in maniera distinta, quindi “Piattaforma” è stata rimossa, e la relazione “Proprietà” è stata sostituita dalle relazioni “Proprietà Store” e “Proprietà Console”



Accorpamenti e partizionamenti:

Dato che l'unico attributo dell'entità "Generazione" è la sola chiave è stato deciso di accorpare l'entità in "Console" aggiungendo il campo "Generazione" a quest'ultima



NOTA: questa modifica impatta molto sull'operazione n. 36 che è stata prima discussa nell'ambito dell'eliminazione delle ridondanze. Con questa ulteriore modifica ora l'operazione richiede i seguenti accessi:

| Operazione 36 | | | |
|---------------|-----------|---------|------|
| Concetto | Costrutto | Accessi | Tipo |
| Console | Entità | 30 | L |

Con, di conseguenza, i costi qua riportati:

$$\text{Op.36} = 30(\text{costo}) * 10(\text{volte al giorno}) = 300$$

$$\text{Totale} = 300$$

Eliminazione degli attributi multivalore:

In questa fase della progettazione non sono state effettuate modifiche in quanto non sono presenti attributi multivalore, nonostante ciò abbiamo è stato preferito comunque inserire questa sezione per completezza di progettazione.

Elenco degli identificatori principali:

| Nome entità | Identificatore |
|-------------|---------------------------------|
| Categoria | Nome (stringa) |
| Utente | Nickname (stringa) |
| Designer | Id (numerico) |
| Prodotto | Nome (stringa), Anno (numerico) |
| Compagnia | Nome (stringa) |
| Saga | Nome (stringa) |
| Store PC | Nome (stringa) |
| Console | Nome (stringa) |

3c. Normalizzazione

Associazioni:

Analizzando lo schema concettuale ristrutturato si nota che tutte le associazioni presenti sono in forma normale di Boyce e Codd perché binarie.

Entità:

| Nome entità | Commento |
|-------------|---|
| Categoria | Non esistono dipendenze non banali tra gli attributi. |
| Utente | Non esistono dipendenze non banali tra gli attributi. |
| Prodotto | Non esistono dipendenze non banali tra gli attributi. L'unica dipendenza è tra descrizione e prezzo che dipendono da nome e anno (banale) |
| Gioco | Non esistono dipendenze non banali tra gli attributi. |
| Designer | Non esistono dipendenze non banali tra gli attributi. L'unica dipendenza presente è tra nome, cognome e nazionalità che dipendono da id (banale) |
| Compagnia | Non esistono dipendenze non banali tra gli attributi. L'unica dipendenza presente è tra fondazione e nazionalità che dipendono dalla Nome (banale) |
| Store PC | Non esistono dipendenze non banali tra gli attributi. |
| Console | Non esistono dipendenze non banali tra gli attributi. L'unica dipendenza presente è tra data di uscita e supporto che dipendono dal nome (banale) |
| Saga | Non esistono dipendenze non banali tra gli attributi. L'unica dipendenza presente è tra descrizione e stato che dipendono dal nome (banale) |

3d. Traduzione verso il modello relazionale

| Entità - Relazione | Traduzione |
|--------------------|---|
| Categoria | Categoria(<u>Nome</u> , derivazione) |
| Utente | Utente(<u>Nickname</u>) |
| Prodotto | Prodotto(<u>Nome</u> , <u>Anno</u> , prezzo, descrizione, produttore, sviluppatore, gioco, saga) |
| Designer | Designer(<u>id</u> , nome, cognome, nazionalità) |
| Compagnia | Compagnia(<u>Nome</u> , Fondazione, nazionalità) |
| Store PC | Store_PC(<u>Nome</u> , compagnia) |
| Console | Console(<u>Nome</u> , data uscita, supporto, compagnia, generazione) |
| Saga | Saga(<u>Nome</u> , descrizione, Stato) |
| Tipologia | Tipologia(<u>Nome_prodotto</u> , <u>anno</u> , <u>Nome_categoria</u>) |
| Recensione | Recensione(<u>Nome</u> , <u>anno</u> , <u>Nickname</u> , Voto) |
| Progettazione | Progettazione(<u>Id_designer</u> , <u>Nome</u> , <u>Anno</u>) |
| Disponibilità | Disponibilità(<u>Nome_prodotto</u> , <u>anno</u> , <u>nome_Store</u> , Requisiti) |
| Riproducibilità | Riproducibilità(<u>Nome_Gioco</u> , <u>Anno</u> , <u>Nome_Console</u>) |

| Traduzione | Vincoli di riferimento |
|-----------------|--|
| Categoria | Derivazione → Categoria.nome |
| Utente | - |
| Prodotto | Produttore → Compagnia.nome Sviluppatore → Compagnia.nome Saga → Saga.nome |
| Designer | - |
| Compagnia | - |
| Store PC | Compagnia → compagnia.nome |
| Console | Compagnia → compagnia.nome |
| Saga | - |
| Tipologia | Nome_prodotto → Prodotto.Nome Anno → Prodotto.Anno Nome_categoria → Prodotto.Nome Nome_prodotto → Prodotto.Nome Anno → Prodotto.Anno Nickname → Utente.Nickname |
| Recensione | Id_designer → Designer.Id Nome → Prodotto.Nome Anno → Prodotto.Anno |
| Progettazione | Nome_prodotto → Prodotto.Nome Anno → Prodotto.Anno Nome_Store → Store.nome |
| Disponibilità | Nome_Gioco → Prodotto.nome Anno → Prodotto.Anno Nome_Console → Console.Nome |
| Riproducibilità | |

4. Codifica SQL

4a. Definizione dello schema

```
CREATE DATABASE Videogiochi_e_software;
USE Videogiochi_e_software;
CREATE TABLE Categoria(
    Nome char(20) not null primary key,
    derivazione char(20),
    foreign key (derivazione) references Categoria(Nome)
);
CREATE TABLE Utente(
    Nickname char(20) not null primary key
);
CREATE TABLE Compagnia(
    Nome char(20) not null primary key,
    Fondazione year,
    Nazionalita char(20)
);
CREATE TABLE Saga(
    Nome char(30) not null primary key,
    Descrizione char(140),
    Stato char(20),
    check(Stato = "Terminata" or Stato = "In corso")
);
CREATE TABLE Prodotto(
    Nome char(30) not null,
    Anno year not null,
    Prezzo decimal(5,2) unsigned,
    Descrizione char(140),
    Produttore char(20) not null,
    Sviluppatore char(20) not null,
    Gioco boolean,
    Saga char(30),
    PRIMARY KEY(Nome, Anno),
    check((Saga IS NOT NULL AND Gioco = True) OR (Saga IS NULL)),
    foreign key (Produttore) references Compagnia(Nome),
    foreign key (Sviluppatore) references Compagnia(Nome),
    foreign key (Saga) references Saga(Nome)
);
CREATE TABLE Designer(
    Id integer auto_increment not null primary key,
    Nome char(20),
    Cognome char(20),
    Nazionalita char(20)
);
```

```

CREATE TABLE Store_PC(
    Nome char(20) not null primary key,
    Compagnia char(20) not null,
    foreign key (Compagnia) references Compagnia(Nome)
);
CREATE TABLE Console(
    Nome char(20) not null primary key,
    Data_Uscita date,
    Supporto bool,
    Compagnia char(20) not null,
    Generazione integer not null,
    foreign key (Compagnia) references Compagnia(Nome)
);
CREATE TABLE Tipologia(
    Nome_prodotto char(30) not null,
    Anno year not null,
    Nome_Categoria char(20) not null,
    PRIMARY KEY(Nome_prodotto, Anno, Nome_Categoria),
    foreign key (Nome_prodotto, Anno) references Prodotto(Nome, Anno),
    foreign key (Nome_Categoria) references Categoria(Nome)
);
CREATE TABLE Recensione(
    Nome char(30) not null,
    Anno year not null,
    Utente char(20) not null,
    Voto tinyint,
    PRIMARY KEY(Nome, Anno, Utente),
    check(voto > 0 and voto < 6),
    foreign key (Nome, Anno) references Prodotto(Nome, Anno),
    foreign key (Utente) references Utente(Nickname)
);
CREATE TABLE Progettazione(
    Id_designer integer not null,
    Nome char(30) not null,
    Anno year not null,
    PRIMARY KEY(Id_designer, Nome, Anno),
    foreign key (Id_designer) references Designer(Id),
    foreign key (Nome, Anno) references Prodotto(Nome, Anno)
);
CREATE TABLE Disponibilita(
    Nome_prodotto char(30) not null,
    Anno year not null,
    Nome_store char(20) not null,
    Processore char(20),
    Scheda_Grafica char(20),
    Os char(20),
    PRIMARY KEY(Nome_prodotto, Anno, Nome_store),

```



```

        foreign key (Nome_prodotto, Anno) references Prodotto(Nome, Anno),
        foreign key (Nome_store) references Store_PC(Nome)
    );
CREATE TABLE Riproducibilita(
    Nome_Gioco char(30) not null,
    Anno year not null,
    Nome_Console char(20) not null,
    PRIMARY KEY(Nome_Gioco, Anno, Nome_Console),
    foreign key (Nome_Gioco, Anno) references Prodotto(Nome, Anno),
    foreign key (Nome_Console) references Console(Nome)
);

```

4b. Creazione della vista

È stato deciso di creare una vista dato che ogni prodotto vuole essere accompagnato dal suo voto, quindi la vista può semplificare le query che coinvolgono i prodotti.

```

CREATE VIEW ProdRecen AS
SELECT
    P.Nome AS Nome,
    P.Anno AS Anno,
    P.Prezzo AS Prezzo,
    P.Descrizione AS Descrizione,
    P.Produttore AS Produttore,
    P.Sviluppatore AS Sviluppatore,
    P.Gioco AS Gioco,
    P.Saga AS Saga,
    AVG(R.Voto) AS Valutazione
FROM Prodotto AS P LEFT JOIN Recensione AS R
    ON P.Nome = R.Nome AND P.Anno = R.Anno
GROUP BY P.Nome, P.Anno, P.Prezzo, P.Descrizione, P.Produttore,
    P.Sviluppatore, P.Gioco, P.Saga;

```

4c. Codifica delle operazioni

Inserire una categoria

```

INSERT INTO Categoria (Nome, derivazione) VALUES (... , ...);
INSERT INTO Tipologia (Nome_prodotto, Anno, Nome_Categoria) VALUES (... , ... , ...);

```

Inserire una compagnia

```

INSERT INTO Compagnia (Nome, Fondazione, Nazionalita) VALUES (... , ... , ...);

```

Inserire una console

```
INSERT INTO Console (Nome, Data_Uscita, Supporto, Compagnia, Generazione)
VALUES (... , ... , ... , ... , ...);
```

Inserire un designer

```
INSERT INTO Designer (Nome, Cognome, Nazionalita) VALUES (... , ... , ...);
```

Inserire la disponibilità di un prodotto per uno store PC

```
INSERT INTO Disponibilita (Nome_prodotto, Anno, Nome_store, Processore, Scheda_Grafica,
Os) VALUES (... , ... , ... , ... , ... , ...);
```

Inserire la riproducibilità di un gioco per una console

```
INSERT INTO Riproducibilita (Nome_Gioco, Anno, Nome_Console) VALUES (... , ... , ...);
```

Inserire un software e la sua categoria d'appartenenza

```
INSERT INTO Prodotto (Nome, Anno, Prezzo, Descrizione, Produttore, Sviluppatore, Gioco)
VALUES (... , ... , ... , ... , ... , ... , '0');
INSERT INTO Tipologia (Nome_prodotto, Anno, Nome_Categoria) VALUES (... , ... , ...);
```

Inserire un gioco con designer che ci hanno lavorato, categorie e saghe di appartenenza

```
INSERT INTO Prodotto (Nome, Anno, Prezzo, Descrizione, Produttore, Sviluppatore, Gioco,
Saga) VALUES (... , ... , ... , ... , ... , ... , '1', ...);
INSERT INTO Progettazione (Id_designer, Nome, Anno) VALUES (... , ... , ...);
INSERT INTO Tipologia (Nome_prodotto, Anno, Nome_Categoria) VALUES (... , ... , ...);
```

Inserire una recensione

```
INSERT INTO Recensione (Nome, Anno, Utente, Voto) VALUES (... , ... , ... , ...);
```

Inserire una saga

```
INSERT INTO `Videogiochi_e_software`.`Saga` (`Nome`, `Descrizione`, `Stato`)
VALUES (... , ... , ...);
UPDATE `Videogiochi_e_software`.`Prodotto` SET `Saga` = 'Super Mario'
WHERE (`Nome` = ...) and (`Anno` = ...);
```

Inserire uno store

```
INSERT INTO `Videogiochi_e_software`.`Store_PC` (`Nome`, `Compagnia`)
VALUES (... , ... , ...);
```

Inserire un utente

```
INSERT INTO Utente (Nickname) VALUES (...);
```

Visualizzare tutte le categorie

```
SELECT Nome FROM Categoria;
```

Visualizzare tutti i prodotti di una categoria

```
SELECT Nome_Prodotto AS Nome, Anno, Valutazione
FROM Categoria AS Cat NATURAL JOIN (
  SELECT Tipo.Nome_Categoria AS Nome, Prod.Nome AS Nome_prodotto, Prod.Anno
  AS Anno, Prod.Valutazione AS Valutazione
  FROM Tipologia AS Tipo INNER JOIN ProdRecen AS Prod
  ON Tipo.Nome_prodotto = Prod.Nome AND Tipo.Anno = Prod.Anno) AS Prodotti
WHERE Nome=...;
```

Visualizzare tutte le compagnie

```
SELECT * FROM Compagnia;
```

Visualizzare tutte le console

```
SELECT * FROM Console;
```

Visualizzare tutti i designer

```
SELECT * FROM Designer;
```

Visualizzare i giochi di una console ordinati per valutazione

```
SELECT Giochi.Nome AS Nome, Giochi.Anno AS Anno, Giochi.Valutazione AS Valutazione
FROM Riproducibilita AS Rip INNER JOIN ProdRecen AS Giochi
  ON Rip.Nome_Gioco = Giochi.Nome AND Rip.Anno = Giochi.Anno
WHERE Giochi.Gioco IS TRUE AND Rip.Nome_Console = ...
ORDER BY Giochi.Valutazione DESC;
```

Visualizzare i giochi di uno store

```
SELECT Giochi.Nome AS Nome, Giochi.Anno AS Anno, Giochi.Valutazione AS Valutazione
FROM Disponibilita AS Disp INNER JOIN ProdRecen AS Giochi
  ON Disp.Nome_prodotto = Giochi.Nome AND Disp.Anno = Giochi.Anno
WHERE Giochi.Gioco IS TRUE AND Disp.Nome_Store= ...;
```

Visualizzare tutti i prodotti

```
SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Valutazione  
FROM ProdRecen;
```

Visualizzare tutti i giochi ordinati per valutazione

```
SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Saga, Valutazione  
FROM ProdRecen  
WHERE Gioco IS TRUE  
ORDER BY Valutazione DESC;
```

Visualizzare tutti i software

```
SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Saga, Valutazione  
FROM ProdRecen  
WHERE Gioco IS FALSE;
```

Visualizzare tutti i designer che hanno lavorato ad un gioco

```
SELECT Des.Nome AS Nome, Des.Cognome AS Cognome, Des.Nazionalita AS Nazionalita  
FROM Designer AS Des INNER JOIN (  
    SELECT id_designer  
    FROM Progettazione NATURAL JOIN Prodotto  
    WHERE Nome=... AND Anno=...  
) AS Id ON Des.Id = Id.Id_designer;
```

Visualizzare tutte le saghe ordinate per anno di nascita

```
SELECT S.Nome AS Nome, S.Descrizione AS Descrizione, S.Stato AS Stato,  
    MIN(P.Anno) AS Nascita  
FROM Saga as S, Prodotto AS P  
WHERE P.Saga = S.Nome  
GROUP BY S.Nome  
ORDER BY MIN(P.Anno);
```

Visualizzare tutti i giochi di una saga

```
SELECT P.Nome AS Nome, P.Anno AS Anno, P.Prezzo AS Prezzo,  
    P.Descrizione AS Descrizione, P.Valutazione as Valutazione  
FROM ProdRecen AS P INNER JOIN Saga AS S ON P.Saga = S.Nome  
WHERE S.Nome=... AND P.Gioco IS TRUE;
```

Visualizzare tutti i produttori

```
SELECT DISTINCT C.Nome AS Compagnie  
FROM Compagnia AS C INNER JOIN Prodotto AS P ON C.Nome = P.Produttore;
```

Visualizzare tutti gli sviluppatori

```
SELECT DISTINCT C.Nome AS Compagnie  
FROM Compagnia AS C INNER JOIN Prodotto AS P ON C.Nome = P.Sviluppatore;
```

Visualizzare tutti gli store di una compagnia

```
SELECT S.Nome AS Store  
FROM Compagnia AS C INNER JOIN Store_PC AS S ON C.Nome = S.Compagnia  
WHERE C.Nome = ...;
```

Visualizzare tutte le console di una compagnia

```
SELECT Con.Nome AS Console  
FROM Compagnia AS C INNER JOIN Console AS Con ON C.Nome = Con.Compagnia  
WHERE C.Nome = ...;
```

Visualizzare tutte le recensioni di un utente

```
SELECT Nome AS Gioco, Anno, Voto  
FROM Recensione  
WHERE Utente = ...;
```

Aggiornare il voto di una recensione

```
UPDATE Recensione SET Voto = ... WHERE Nome = ... AND Anno = ... AND Utente = ...;
```

Aggiornare il prezzo di un gioco

```
UPDATE Prodotto SET Prezzo = ... WHERE Nome = ... AND Anno = ...;
```

Aggiornare la saga di un gioco

```
UPDATE Prodotto SET Saga = ... WHERE Nome = ... AND Anno = ...;
```

Visualizzare tutti i sottogeneri di una categoria

```
WITH RECURSIVE Principale(Derivato, Derivante) AS  
(  
    SELECT *  
    FROM Categoria  
    UNION ALL  
    SELECT Nome, Derivante
```

```

        FROM Principale, Categoria
        WHERE Derivato = derivazione
    )
    SELECT Derivato as Categorie
    FROM Principale
    WHERE Derivante = ...;

```

Visualizzare tutte le compagnie per cui ha lavorato un designer

```

SELECT DISTINCT Compagnia.Nome AS Nome, Compagnia.Fondazione AS Fondazione,
               Compagnia.Nazionalita AS Nazionalita
FROM Designer, Progettazione, Prodotto, Compagnia
WHERE Designer.Id = Progettazione.Id_designer AND Progettazione.Nome = Prodotto.Nome
AND   Progettazione.Anno = Prodotto.Anno AND Prodotto.Produttore = Compagnia.Nome AND
      Prodotto.Sviluppatore = Compagnia.Nome AND Designer.Id = ...;

```

Visualizzare tutte le generazioni supportate con la loro data di nascita

```

SELECT Generazione, MIN(Data_Uscita) AS Inizio
FROM Console
WHERE Supporto IS TRUE
GROUP BY Generazione;

```

Visualizzare tutti i software di un'azienda ordinati per prezzo

```

SELECT Prod.Nome AS Nome, Prod.Prezzo AS Prezzo
FROM Prodotto AS Prod INNER JOIN Compagnia AS Comp
      ON Prod.Produttore = Comp.Nome
WHERE Prod.Gioco IS FALSE AND Comp.Nome=...
ORDER BY Prod.Prezzo;

```

Visualizzare chi ha prodotto i prodotti sviluppati da una compagnia

```

SELECT DISTINCT Prod.Produttore AS Produttori
FROM Compagnia AS Comp INNER JOIN Prodotto AS Prod
      ON Comp.Nome = Prod.Produttore
WHERE Prod.Sviluppatore = ... AND Prod.Gioco IS TRUE;

```

Aggiungere una nuova tipologia ad un prodotto

```

INSERT INTO Tipologia (Nome_prodotto, Anno, Nome_Categoria) VALUES (... , ... , ...);

```

Aggiornare la lista di designer che hanno lavorato ad un gioco

```

INSERT INTO Progettazione (Id_designer, Nome, Anno) VALUES (... , ... , ...);

```

Aggiornare il supporto di una determinata console

UPDATE Console SET Supporto = ... WHERE Nome = ...;

5. Testing

All'indirizzo github.com/SeppiaBrilla/Progetto_Base_di_Dati è possibile trovare, nella sotto cartella "Database" lo script sql che permette la creazione del database su mysql, gli script che, se importati nel database, lo popolano come da testing del 24/12/2020 ed infine script che eseguono le queries identificate nella sezione operazioni e codificate nella sezione 4c.

All'indirizzo site192026.tw.cs.unibo.it si può invece trovare un esempio di sito web che permette di eseguire le operazioni sul database da interfaccia grafica. Per provarlo basta scegliere un'operazione dal menù a tendina, inserire i dati nel caso fossero necessari (es. Per inserimenti o operazioni su dati specifici) e premere il pulsante che rimanderà ad una pagina con i risultati. Di seguito si allegano screenshot che mostrano i risultati delle operazioni di lettura (dalla 13 alla 30 e dalla 34 alla 38) con dati d'esempio nel caso fosse necessario

13

Nome

Action

Avventura

Musica

Platform

RPG

Shooting

Sport

Survival

Utility

SJRPG

JRPG

MMORPG

FPS

TPS

E' stata eseguita la query: SELECT Nome FROM Categoria;

14

Nome

Anno

Valutazione

Crash Bandicoot

1999

E.T.

1975

Super Mario Odissey

2018

E' stata eseguita la query: SELECT Nome_Prodotto AS Nome, Anno, Valutazione FROM Categoria AS Cat NATURAL JOIN (SELECT Tipo.Nome_Categoria AS Nome, Prod.Nome AS Nome_prodotto, Prod.Anno AS ANNO, Prod.Valutazione AS Valutazione FROM Tipologia AS Tipo INNER JOIN ProdRecen AS Prod ON Tipo.Nome_prodotto = Prod.Nome AND Tipo.Anno = Prod.Anno) AS Prodotti WHERE Nome='Platform';

15

| Nome | Fondazione | Nazionalita |
|---|------------|-------------|
| Apple | 1986 | USA |
| Atari | 1971 | USA |
| CD Project Red | 1996 | Polonia |
| Intelligent Systems | 1976 | Giappone |
| Microsoft | 1984 | USA |
| Nintendo | 1983 | Giappone |
| Sony | 1946 | Giappone |
| Square Soft | 1985 | Giappone |
| Studio MDHR | 2000 | USA |
| Ubisoft | 1990 | Francia |
| Valve | 1991 | USA |
| È stata eseguita la query: SELECT * FROM Compagnia; | | |

16

| Nome | Data_Uscita | Supporto | Compagnia | Generazione |
|---|---|----------|-----------|-------------|
| Atari 2600 | Sun Sep 11 1977 00:00:00 GMT+0000 (UTC) | 0 | Atari | 2 |
| Color TV Game | Sun Dec 25 1977 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 1 |
| Gamecube | Sun Nov 18 2001 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 6 |
| NES | Fri Oct 18 1985 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 3 |
| Nintendo 64 | Sun Sep 29 1996 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 5 |
| Playstation 1 | Sat Sep 09 1995 00:00:00 GMT+0000 (UTC) | 0 | Sony | 5 |
| Playstation 2 | Mon Mar 01 1999 00:00:00 GMT+0000 (UTC) | 0 | Sony | 6 |
| Playstation 3 | Fri Nov 17 2006 00:00:00 GMT+0000 (UTC) | 0 | Sony | 7 |
| Playstation 4 | Fri Nov 15 2013 00:00:00 GMT+0000 (UTC) | 1 | Sony | 8 |
| SNES | Mon Aug 12 1991 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 4 |
| Switch | Fri Mar 03 2017 00:00:00 GMT+0000 (UTC) | 1 | Nintendo | 8 |
| Wii | Sun Nov 19 2006 00:00:00 GMT+0000 (UTC) | 0 | Nintendo | 7 |
| È stata eseguita la query: SELECT * FROM Console; | | | | |

17

| Id | Nome | Cognome | Nazionalita |
|--|--------------|------------|-------------|
| 1 | Shigeru | Miyamoto | Giappone |
| 2 | Jason | Rubin | USA |
| 3 | Jared | Moldenauer | USA |
| 4 | Chad | Moldenauer | USA |
| 5 | Michael Alyn | Pondsmith | Polonia |
| 6 | Yuji | Ori | Giappone |
| 7 | Atari | Guy | USA |
| 8 | Yuji | Ohashi | Giappone |
| È stata eseguita la query: SELECT * FROM Designer; | | | |

18

| Nome | Anno | Valutazione |
|--|------|-------------|
| Cuphead | 2018 | |
| Super Mario Odyssey | 2018 | |
| È stata eseguita la query: SELECT Giochi.Nome AS Nome, Giochi.Anno AS Anno, Giochi.Valutazione AS Valutazione FROM Riproducibilita AS Rip INNER JOIN ProdRecen AS Giochi ON Rip.Nome_Gioco = Giochi.Nome AND Rip.Anno = Giochi.Anno WHERE Giochi.Gioco IS TRUE AND Rip.Nome_Console = 'Switch' ORDER BY Giochi.Valutazione DESC; | | |
| Wii Sport | 2006 | 0 |
| Crash Bandicoot | 1999 | 49.99 |
| Crash Nitro Kart | 2004 | 36.99 |
| Cuphead | 2018 | 19.99 |
| E.T. | 1975 | 39.99 |
| Final Fantasy 2 | 1980 | 49.99 |
| Fire Emblem Awakening | 2006 | 39.9 |
| Super Mario Odyssey | 2018 | 69.99 |
| Zelda: A Link To The Past | 1989 | 45.99 |
| Zelda: Twilight Princess | 2003 | 39.99 |
| È stata eseguita la query: SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Saga, Valutazione FROM ProdRecen WHERE Gioco IS TRUE ORDER BY Valutazione DESC; | | |

19

| Nome | Anno | Valutazione |
|----------------|------|-------------|
| Cuphead | 2018 | |
| Cyberpunk 2077 | 2020 | 3.5 |

È stata eseguita la query: SELECT Giochi.Nome AS Nome, Giochi.Anno AS Anno, Giochi.Valutazione AS Valutazione FROM Disponibilita AS Disp INNER JOIN ProdRecen AS Giochi ON Disp.Nome_prodotto = Giochi.Nome AND Disp.Anno = Giochi.Anno WHERE Giochi.Gioco IS TRUE AND Disp.Nome_Store= 'Steam';

20

| Nome | Anno | Descrizione | Produttore | Sviluppatore | Valutazione |
|---------------------------|------|-------------|----------------|---------------------|-------------|
| Crash Bandicoot | 1999 | 49.99 | Sony | Sony | |
| Crash Nitro Kart | 2004 | 36.99 | Sony | Sony | |
| Cuphead | 2018 | 19.99 | Microsoft | Studio MDHR | |
| Cyberpunk 2077 | 2020 | 59.99 | CD Project Red | CD Project Red | 3.5 |
| E.T. | 1975 | 39.99 | Atari | Atari | |
| Final Fantasy 2 | 1980 | 49.99 | Square Soft | Square Soft | |
| Fire Emblem Awakening | 2006 | 39.9 | Nintendo | Intelligent Systems | |
| Garage Band | 2008 | 0 | Apple | Apple | |
| Goldeneye | 1996 | 39.99 | Nintendo | Nintendo | 3 |
| OneNote | 2017 | 0 | Microsoft | Microsoft | 1 |
| Outlook | 2017 | 0 | Microsoft | Microsoft | |
| Survox | 2006 | 0 | Microsoft | Microsoft | |
| Super Mario Odyssey | 2018 | 69.99 | Nintendo | Nintendo | |
| Tennis A | 1971 | 0 | Nintendo | Nintendo | 1 |
| The Last Of Us | 2012 | 60 | Sony | Sony | 2 |
| Wii Sport | 2006 | 0 | Nintendo | Nintendo | 1 |
| Zelda: A Link To The Past | 1989 | 45.99 | Nintendo | Nintendo | |
| Zelda: Twilight Princess | 2003 | 39.99 | Nintendo | Nintendo | |

È stata eseguita la query: SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Valutazione FROM ProdRecen;

21

| Nome | Anno | Descrizione | Produttore | Sviluppatore | Saga | Valutazione |
|---------------------------|------|-------------|----------------|---------------------|---------------------|-------------|
| Cyberpunk 2077 | 2020 | 59.99 | CD Project Red | CD Project Red | | 3.5 |
| Goldeneye | 1996 | 39.99 | Nintendo | Nintendo | | 3 |
| The Last Of Us | 2012 | 60 | Sony | Sony | The Last Of Us | 2 |
| Tennis A | 1971 | 0 | Nintendo | Nintendo | | 1 |
| Wii Sport | 2006 | 0 | Nintendo | Nintendo | | 1 |
| Crash Bandicoot | 1999 | 49.99 | Sony | Sony | Crash Bandicoot | |
| Crash Nitro Kart | 2004 | 36.99 | Sony | Sony | Crash Bandicoot | |
| Cuphead | 2018 | 19.99 | Microsoft | Studio MDHR | | |
| E.T. | 1975 | 39.99 | Atari | Atari | | |
| Final Fantasy 2 | 1980 | 49.99 | Square Soft | Square Soft | | |
| Fire Emblem Awakening | 2006 | 39.9 | Nintendo | Intelligent Systems | | |
| Super Mario Odyssey | 2018 | 69.99 | Nintendo | Nintendo | | |
| Zelda: A Link To The Past | 1989 | 45.99 | Nintendo | Nintendo | The Legend Of Zelda | |
| Zelda: Twilight Princess | 2003 | 39.99 | Nintendo | Nintendo | The Legend Of Zelda | |

È stata eseguita la query: SELECT Nome, Anno, Prezzo Descrizione, Produttore, Sviluppatore, Saga, Valutazione FROM ProdRecen WHERE Gioco IS TRUE ORDER BY Valutazione DESC;

22

| Nome | Anno | Descrizione | Produttore | Sviluppatore | Saga | Valutazione |
|-------------|------|-------------|------------|--------------|------|-------------|
| Garage Band | 2008 | 0 | Apple | Apple | | |
| OneNote | 2017 | 0 | Microsoft | Microsoft | | 1 |
| Outlook | 2017 | 0 | Microsoft | Microsoft | | |
| Survox | 2006 | 0 | Microsoft | Microsoft | | |

23

| Nome | Cognome | Nazionalita |
|-------|------------|-------------|
| Jared | Moldenauer | USA |
| Chad | Moldenauer | USA |

È stata eseguita la query: SELECT Des.Nome AS Nome, Des.Cognome AS Cognome, Des.Nazionalita AS Nazionalita FROM Designer AS Des INNER JOIN (SELECT id_designer FROM Progettazione NATURAL JOIN Prodotto WHERE Nome='Cuphead' AND Anno='2018') AS Id ON Des.Id = Id.Id_designer;

24

| Nome | Descrizione | Stato | Nascita |
|---------------------|--|-----------|---------|
| The Legend Of Zelda | Link, your princess is in another castle | In Corso | 1989 |
| Crash Bandicoot | Cos'è un bandicoot? Ed è davvero terminato??? | Terminata | 1999 |
| The Last Of Us | Zombie che uccidono i tizi ma alla fine uccidi te i tizi | In Corso | 2012 |

È stata eseguita la query: SELECT S.Nome AS Nome, S.Descrizione AS Descrizione, S.Stato AS Stato, MIN(P.Anno) AS Nascita FROM Saga as S, Prodotto AS P WHERE P.Saga = S.Nome GROUP BY S.Nome ORDER BY MIN(P.Anno);

25

| Nome | Anno | Prezzo | Descrizione | Valutazione |
|---------------------------|------|--------|--------------------------------|-------------|
| Zelda: A Link To The Past | 1989 | 45.99 | Link suona l'ocarina anche qui | |
| Zelda: Twilight Princess | 2003 | 39.99 | Link è un lupo: 10/10 | |

È stata eseguita la query: SELECT P.Nome AS Nome, P.Anno AS Anno, P.Prezzo AS Prezzo, P.Descrizione AS Descrizione, P.Valutazione as Valutazione FROM ProdRecen AS P INNER JOIN Saga AS S ON P.Saga = S.Nome WHERE S.Nome='The Legend of Zelda' AND P.Gioco IS TRUE;

26

| Compagnie |
|----------------|
| Apple |
| Atari |
| CD Project Red |
| Microsoft |
| Nintendo |
| Sony |
| Square Soft |

È stata eseguita la query: SELECT DISTINCT C.Nome AS Compagnie FROM Compagnia AS C INNER JOIN Prodotto AS P ON C.Nome = P.Produttore;

27

| Compagnie |
|---------------------|
| Apple |
| Atari |
| CD Project Red |
| Intelligent Systems |
| Microsoft |
| Nintendo |
| Sony |
| Square Soft |
| Studio MDHR |

È stata eseguita la query: SELECT DISTINCT C.Nome AS Compagnie FROM Compagnia AS C INNER JOIN Prodotto AS P ON C.Nome = P.Sviluppatore;

28

| Store |
|-------|
| Steam |

È stata eseguita la query: SELECT S.Nome AS Store FROM Compagnia AS C INNER JOIN Store_PC AS S ON C.Nome = S.Compagnia WHERE C.Nome = 'Valve';

29

| Console |
|---------------|
| Playstation 1 |
| Playstation 2 |
| Playstation 3 |
| Playstation 4 |

È stata eseguita la query: SELECT Con.Nome AS Console FROM Compagnia AS C INNER JOIN Console AS Con ON C.Nome = Con.Compagnia WHERE C.Nome = 'Sony';

30

| Gioco | Anno | Voto |
|----------|------|------|
| OneNote | 2017 | 1 |
| Tennis A | 1971 | 1 |

È stata eseguita la query: SELECT Nome AS Gioco, Anno, Voto FROM Recensione WHERE Utente = 'SeppiaBrilla';

34

35

| Nome | Fondazione | Nazionalita |
|----------|------------|-------------|
| Nintendo | 1983 | Giappone |

È stata eseguita la query: SELECT DISTINCT Compagnia.Nome AS Nome, Compagnia.Fondazione AS Fondazione, Compagnia.Nazionalita AS Nazionalita FROM Designer, Progettazione, Prodotto, Compagnia WHERE Designer.Id = Progettazione.Id_designer AND Progettazione.Nome = Prodotto.Nome AND Progettazione.Anno = Prodotto.Anno AND Prodotto.Produttore = Compagnia.Nome AND Prodotto.Sviluppatore = Compagnia.Nome AND Designer.Id = 1;

| | | |
|----|--|---|
| 36 | Generazione | Inizio |
| | 8 | Fri Nov 15 2013 00:00:00 GMT+0000 (UTC) |
| | È stata eseguita la query: SELECT Generazione, MIN(Data_Uscita) AS Inizio FROM Console WHERE Supporto IS TRUE GROUP BY Generazione; | |
| 37 | Nome | Prezzo |
| | OneNote | 0 |
| | Outlook | 0 |
| | Survox | 0 |
| | È stata eseguita la query: SELECT Prod.Nome AS Nome, Prod.Prezzo AS Prezzo FROM Prodotto AS Prod INNER JOIN Compagnia AS Comp ON Prod.Produttore = Comp.Nome WHERE Prod.Gioco IS FALSE AND Comp.Nome='Microsoft' ORDER BY Prod.Prezzo; | |
| 38 | Produttori | |
| | Nintendo | |
| | È stata eseguita la query: SELECT DISTINCT Prod.Produttore AS Produttori FROM Compagnia AS Comp INNER JOIN Prodotto AS Prod ON Comp.Nome = Prod.Produttore WHERE Prod.Sviluppatore = 'Intelligent systems' AND Prod.Gioco IS TRUE; | |