



**Univerità degli Studi di Salerno**

Dipartimento di Ingegneria dell'Informazione ed Elettrica e Matematica  
Applicata | Corso di Laurea in Ingegneria Informatica

## **Traccia 6 - Sentiment Analysis: La Gestione Delle Review**

### **Authors**

Lungarella Raffaele	0612705221
Marmo Alberto	0612705220
Martelli Marco	0612705075
Massera Artini Stefano	0612705217

**Anno accademico 2022-2023**

# Contents

<b>1</b>	<b>Descrizione della realtà di interesse</b>	<b>4</b>
1.1	Analisi della realtà di interesse . . . . .	4
<b>2</b>	<b>Analisi delle specifiche</b>	<b>6</b>
2.1	Glossario dei Termini . . . . .	6
2.2	Strutturazione dei requisiti in frasi . . . . .	7
2.2.1	Frasi di carattere generale . . . . .	7
2.2.2	Frasi relative a <u>recensione</u> . . . . .	7
2.2.3	Frasi relative a <u>studio</u> . . . . .	7
2.2.4	Frasi relative ad <u>artwork</u> . . . . .	7
2.2.5	Frasi relative a <u>utente</u> . . . . .	7
2.2.6	Frasi relative ad <u>artista</u> . . . . .	7
2.2.7	Frasi relative a <u>commento</u> . . . . .	8
2.3	Identificazione delle operazioni principali . . . . .	8
<b>3</b>	<b>Progettazione Concettuale</b>	<b>9</b>
3.1	Schema Concettuale . . . . .	9
3.1.1	Note sullo schema E-R . . . . .	9
3.2	Design Pattern . . . . .	10
3.2.1	Pattern "Parte di" . . . . .	10
3.2.2	Pattern "Generalizzazione" . . . . .	11
3.3	Dizionario dei dati . . . . .	12
3.4	Regole Aziendali . . . . .	15
<b>4</b>	<b>Progettazione Logica</b>	<b>16</b>
4.1	Analisi delle Prestazioni . . . . .	16
4.2	Eliminazione delle generalizzazioni . . . . .	19
4.2.1	Generalizzazione "Utente" . . . . .	19
4.2.2	Generalizzazione "Impiego" . . . . .	21
4.3	Partizionamento/Accorpamento delle Entità e Associazioni . . . . .	22
4.4	Scelta degli identificatori principali . . . . .	22
4.5	Schema finale ristrutturato . . . . .	24
4.6	Schema logico . . . . .	24
4.7	Documentazione dello schema logico . . . . .	27
<b>5</b>	<b>Normalizzazione</b>	<b>30</b>
5.1	Verifica prima forma normale (1FN) . . . . .	30
5.2	Verifica seconda forma normale (2FN) . . . . .	30
5.3	Verifica terza forma normale (3FN) . . . . .	30
5.4	Verifica forma normale di Boyce-Codd . . . . .	31
<b>6</b>	<b>Query SQL</b>	<b>32</b>
6.1	Query con operatore di aggregazione e join: Classifica impiegati dell'anno . . . . .	32

6.2	Query nidificata complessa: Ricerca artisti versatili . . . . .	33
6.3	Query insiemistica: Arti tradizionali . . . . .	34
6.4	Eventuali Altre query . . . . .	35
6.4.1	Ricerca competenze specifiche . . . . .	35
<b>7</b>	<b>Viste</b>	<b>35</b>
7.1	Vista <i>recensione_studio</i> . . . . .	35
7.1.1	Query con Vista: Impiegato più produttivo sulla piattaforma	36
7.2	Vista <i>stato_studio</i> . . . . .	37
<b>8</b>	<b>Trigger</b>	<b>38</b>
8.1	Trigger inizializzazione . . . . .	38
8.1.1	Trigger: Controllo Recensione Multimediale . . . . .	38
8.1.2	Trigger: Controlla Risposta . . . . .	40
8.1.3	Trigger: Controllo Impiegati Studio . . . . .	41
8.1.4	Trigger: Controllo Artista . . . . .	41
8.1.5	Trigger: Aggiungi Numero Artwork . . . . .	42
8.2	Trigger per vincoli aziendali . . . . .	43
8.2.1	Trigger: Aggiungi Mi Piace Commento . . . . .	43
8.2.2	Trigger: Elimina Mi Piace Commento . . . . .	43
8.2.3	Trigger: Aggiungi Mi Piace Recensione . . . . .	44
8.2.4	Trigger: Elimina Mi Piace Recensione . . . . .	44
8.2.5	Trigger: Aggiorna Mi Piace Studio . . . . .	45
8.2.6	Trigger: Controllo Opera Personale . . . . .	46

# 1 Descrizione della realtà di interesse

La piattaforma ***Artstation*** intende realizzare una suite di applicazioni per la memorizzazione e l'analisi delle recensioni multimediali e delle informazioni relative agli utenti e agli artisti.

- Per tali **recensioni multimediali**, definite come la possibilità da parte dell'utente di esprimere sensazioni, emozioni, sentimenti, opinioni, punti di vista, giudizi, valutazioni, ecc., propri o altrui, tramite l'utilizzo di "contenuti multimediali" come immagini, si intende memorizzare: gli artwork, l'autore, il titolo, la data di caricamento, la descrizione, i commenti, il numero di "mi piace", il numero di visualizzazioni e i tags. Inoltre, i post devono essere categorizzati relativamente al medium utilizzato (arte 3D, arte 2D, pittura tradizionale ...) e alla tipologia dell'opera d'arte (illustrazioni, ritratti, opere architettoniche ...);
- Per gli **artwork** memorizziamo il file multimediale, il suo MIME-type, la risoluzione (orizzontale e verticale), il profilo colore e infine una didascalia;
- Per i **commenti** rappresentiamo il contenuto del testo, l'autore, la recensione al quale fa riferimento, i "mi piace" e la data di pubblicazione.
- Sulla piattaforma è possibile avere utenti base ed utenti artisti.
- Per quanto riguarda gli **utenti** base, memorizziamo le seguenti informazioni: nome, cognome, data di nascita, localizzazione (nazione e città), e-mail, username e password. Gli utenti base hanno la possibilità di mettere "mi piace" alle recensioni multimediali e ai commenti all'interno della piattaforma.
- Per gli **artisti**, oltre alle informazioni possedute dagli utenti base memorizziamo: le recensioni multimediali pubblicate, i commenti inseriti, la professione, le skills, i software che sono in grado di utilizzare e una breve descrizione sulla loro carriera.
- Gli artisti possono essere impiegati in uno studio, ossia un ambiente dove più artisti lavorano per la realizzazione di opere.
- Per quanto riguarda gli **studio** memorizziamo: il nome, il codice fiscale aziendale, la localizzazione dello studio (nazione, città), i suoi contatti (e-mail, sito) ed una descrizione.

## 1.1 Analisi della realtà di interesse

***Artstation*** è una piattaforma online dedicata alla pubblicazione ed alla condivisione di opere d'arte, come disegni, illustrazioni, animazioni e modelli 3D sotto forma di post.

L'obiettivo del progetto è la creazione di un database che raccolga le informazioni relative ai post ed ai commenti ad esse associati.

Gli utenti di Artstation sono formati da professionisti e da freelancer, i quali hanno la possibilità di pubblicare le proprie opere all'interno di post (recensioni multimediali). Oltre al caricamento delle proprie opere d'arte ed alla possibilità di visualizzare le opere di altri artisti nella sezione "Explore", Artstation permette agli artisti di imparare nella sezione "Learn", di vendere le proprie opere d'arte ed i propri corsi nella sezione "Shop", infine permette di trovare lavoro nella sezione "Jobs".

Le sezioni "Learn", "Shop" e "Jobs" non saranno considerate, in quanto non rientrano nel nostro campo di interesse.

Nel nostro progetto, abbiamo deciso di limitare la nostra attenzione a soli due tipi di utenti : "utente base" e "utente artista", anziché analizzare tutte le numerose categorie presenti sulla piattaforma (*non saranno considerati artisti non utenti della piattaforma*).

Gli **utenti base** potranno accedere alla sezione "Explore", **visualizzare** i post e i commenti e **mettere mi piace**. Tuttavia, gli utenti artisti avranno privilegi aggiuntivi rispetto agli utenti base. Oltre a poter fare tutto ciò che gli utenti base possono fare, gli utenti artisti avranno la possibilità di **pubblicare** i propri post nella sezione "Explore" e di **commentare** i post di altri utenti artisti.

Per diventare un **artista** sulla piattaforma, è necessario completare il profilo con informazioni aggiuntive, pubblicare almeno una recensione multimediale e specificare la propria professione e una breve descrizione della loro carriera. L'artista ha la possibilità di aggiungere informazioni facoltative come **skills** e **software** utilizzati, scegliendo da liste predefinite di valori verificati. È possibile inserire più skills e software o non averne nessuno.

Più artisti potranno partecipare a gruppi chiamati **studio**, associazioni presso le quali gli artisti possono pubblicare opere. Terremo traccia della partecipazione degli artisti negli Studio, andando a memorizzare il loro impiego attuale ed eventuali impieghi passati.

Oltre agli artisti il focus del nostro progetto si incentrerà sui loro post, considerati recensioni **puramente multimediali** in quanto descrivono le sensazioni, le emozioni, le opinioni dell'utente attraverso "contenuti multimediali".

Una recensione multimediale può avere più di un **artwork**. Per arricchire ulteriormente le recensioni, gli utenti potranno utilizzare una serie di attributi, tra cui il titolo, la descrizione, le didascalie e altre informazioni utili per categorizzare il post, come la tipologia di opera d'arte, il medium e i relativi tag.

Le **tipologie** ed i **medium** saranno rappresentati da una lista predefinita di valori stabiliti e **verificati**. Ogni recensione dovrà includere almeno una tipologia e un medium, con un limite massimo di 3 tipologie e 3 medium.

I **tag** sono stringhe che vengono utilizzate per identificare in modo più accurato un determinato contenuto, al fine di fornire all'artista una maggiore visibilità.

Soltanto gli artisti avranno la possibilità di aggiungere i tag, e tutti gli artisti potranno utilizzarli per le proprie recensioni.

L'artista avrà modo di ricevere feedback sia da altri utenti tramite i “**mi piace**” sia da altri artisti tramite **mi piace** e **commenti**. I commenti saranno considerati come una recensione non strutturata di ordine inferiore rispetto alla recensione principale.

Una recensione può avere più commenti, a loro volta i commenti possono avere altri commenti che andremo a definire **risposte**. È possibile lasciare più risposte a un commento, ma non è possibile creare una gerarchia di risposte annidate sotto un singolo commento.

## 2 Analisi delle specifiche

### 2.1 Glossario dei Termini

	Termine	Descrizione	Sinonimi	Collegamenti
1	Recensione multimediale	Pubblicazione di un artista.	Pubblicazione, Post	Artwork, Commento, Artista
2	Studio	Ambiente dove più artisti collaborano per la realizzazione di opere		Artista
3	Artwork	Singola realizzazione di un artista.	Opera, File multimediale	Recensione, Artista
4	Utente	Utilizzatore della piattaforma.	Visitatore, Fan	
5	Artista	Utente che pubblica le opere.	Proprietario, Autore	Artwork, Recensione multimediale, Commento, Studio
6	Commento	Recensione di secondo livello, relativa ad un'opera.	Opinione, Valutazione, Giudizio	Recensione, Artista

Table 1: Glossario dei termini

## **2.2 Strutturazione dei requisiti in frasi**

### **2.2.1 Frasi di carattere generale**

La piattaforma *Artstation* intende realizzare una suite di applicazioni per la memorizzazione e l'analisi delle recensioni multimediali e delle informazioni relative agli utenti.

### **2.2.2 Frasi relative a recensione**

Per tali recensioni, definite come la possibilità da parte dell'utente di esprimere sensazioni, emozioni, sentimenti, opinioni, punti di vista, giudizi, valutazioni, ecc., propri o altrui, tramite l'utilizzo di "contenuti multimediali" come immagini, si intende memorizzare: gli artwork, l'autore, il titolo, la data di caricamento, la descrizione, i commenti, il numero di "mi piace", il numero di visualizzazioni e i tags. Inoltre, i post devono essere categorizzati relativamente al medium utilizzato (arte 3D, arte 2D, pittura tradizionale ...) e alla tipologia dell'opera d'arte (illustrazioni, ritratti, opere architettoniche ...);

### **2.2.3 Frasi relative a studio**

Gli artisti possono essere impiegati in uno studio, ossia un ambiente dove più artisti lavorano per la realizzazione di opere.

Per quanto riguarda gli studio memorizziamo: il nome, il codice fiscale aziendale, la localizzazione dello studio (nazione, città), i suoi contatti (e-mail, sito) ed una descrizione.

### **2.2.4 Frasi relative ad artwork**

Per gli artwork memorizziamo il file multimediale, il suo MIME-type, la risoluzione (orizzontale e verticale), il profilo colore e infine una didascalia;

### **2.2.5 Frasi relative a utente**

Per quanto riguarda gli utenti base, memorizziamo le seguenti informazioni: nome, cognome, data di nascita, localizzazione (nazione e città), e-mail, username e password. Gli utenti base hanno la possibilità di mettere "mi piace" alle recensioni multimediali e ai commenti all'interno della piattaforma.

### **2.2.6 Frasi relative ad artista**

Per gli artisti, oltre alle informazioni possedute dagli utenti base memorizziamo: le recensioni multimediali pubblicate, i commenti inseriti, la professione, le skills, i software che sono in grado di utilizzare e una breve descrizione sulla loro carriera.

### **2.2.7 Frasi relative a commento**

Per i commenti rappresentiamo il contenuto del testo, l'autore, la recensione al quale fa riferimento, i "mi piace" e la data di pubblicazione.

## **2.3 Identificazione delle operazioni principali**

**Operazione 1:** pubblicazione di una recensione multimediale (15.000-20.000 volte al giorno)

**Operazione 2:** mostrare tutte le opere personali pubblicate da un determinato artista (80.000-100.000 volte al giorno).

**Operazione 3:** mostrare tutte le opere associate ad artisti afferenti ad un determinato studio (40.000-50.000 volte al giorno).

**Operazione 4:** mostra il primo artwork relativo a tutte le recensioni multimediali che contengono una determinata parola nei campi "Descrizione", "Titolo", "Tag" e nel campo "Didascalia" degli artwork relativi alle recensioni (125.000-150.000 volte al giorno).

**Operazione 5:** acquisizione degli indirizzi e-mail degli utenti per l'aggiornamento della newsletter (1 volta a settimana)

**Operazione 6:** mostrare l'artista di tendenza (artista le cui recensioni hanno ricevuto il maggior numero di "mi piace") del mese (1 volta al mese).



## 3 Progettazione Concettuale

### 3.1 Schema Concettuale

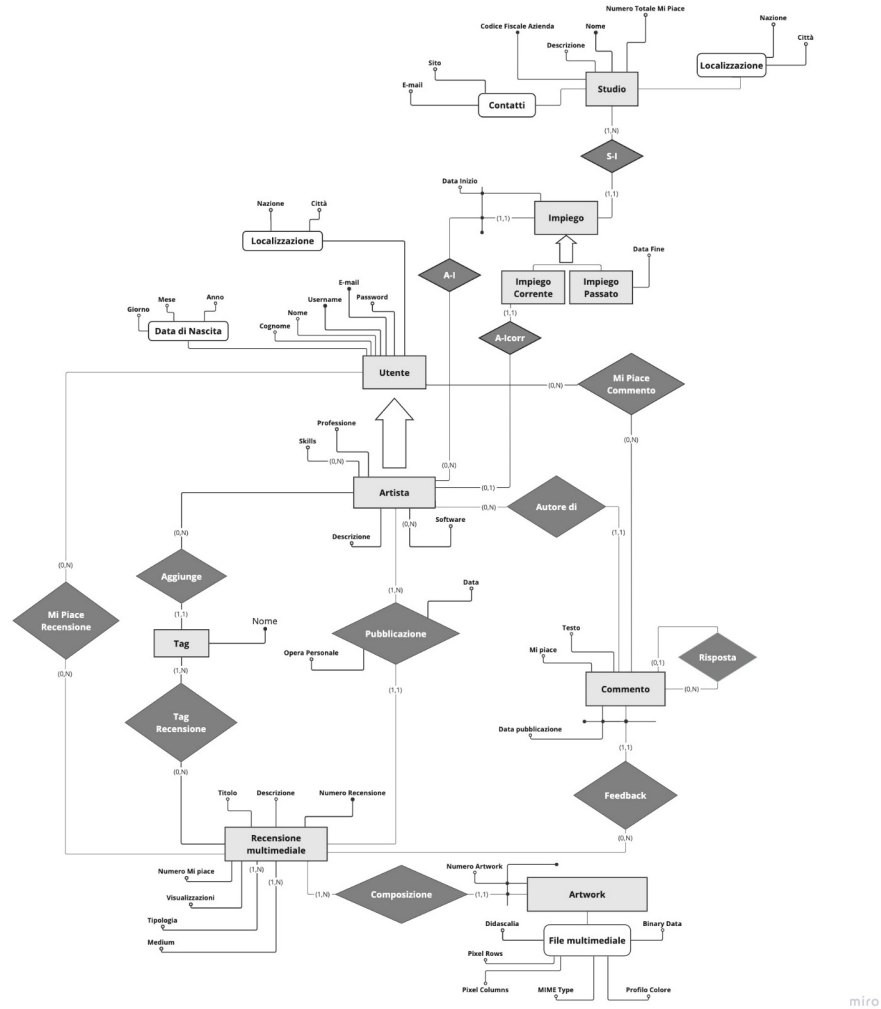


Figure 1: Schema Concettuale

#### 3.1.1 Note sullo schema E-R

L'attributo "Opera Personale" nell'associazione "Pubblicazione" svolge un ruolo chiave per distinguere tra opere personali e opere legate allo studio in cui l'artista è impiegato. È un booleano che indica se un'opera è di natura personale o associata al lavoro nello studio attuale.

## 3.2 Design Pattern

### 3.2.1 Pattern "Parte di"

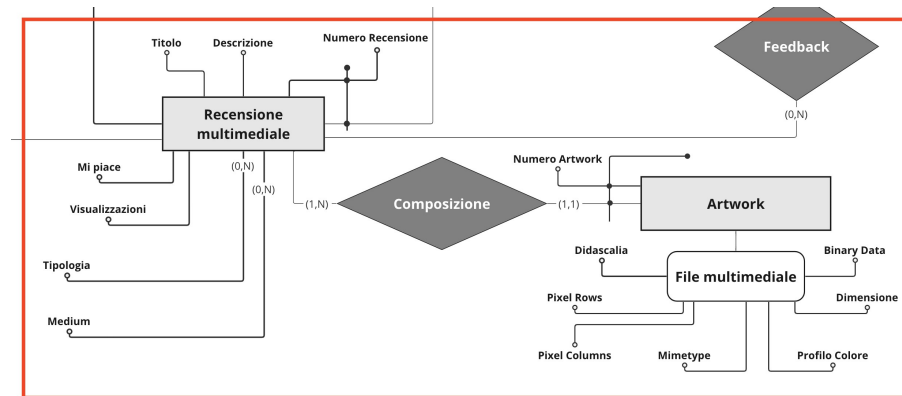


Figure 2: Pattern "Parte di" su Artwork

Tale pattern è stato applicato poiché l'entità Artwork è parte della recensione multimediale. L'esistenza di un'occorrenza dell'entità Artwork dipende dall'esistenza di un'occorrenza di una recensione multimediale e richiede l'identificazione esterna.

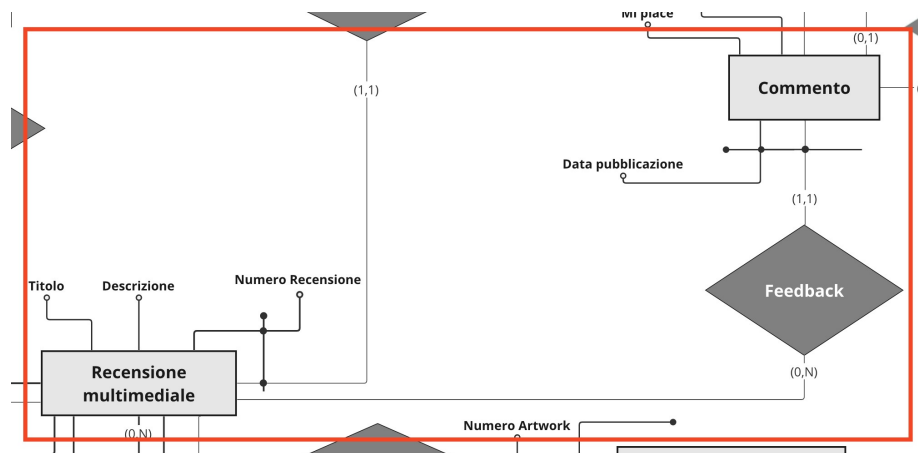


Figure 3: Pattern "Parte di" su Commento

Anche in questo caso si è scelto di applicare il pattern "Parte di" tra le entità Recensione multimediale e Commento. Infatti, quest'ultimo dipende dall'esistenza dell'entità recensione multimediale e richiede un'identificazione esterna.

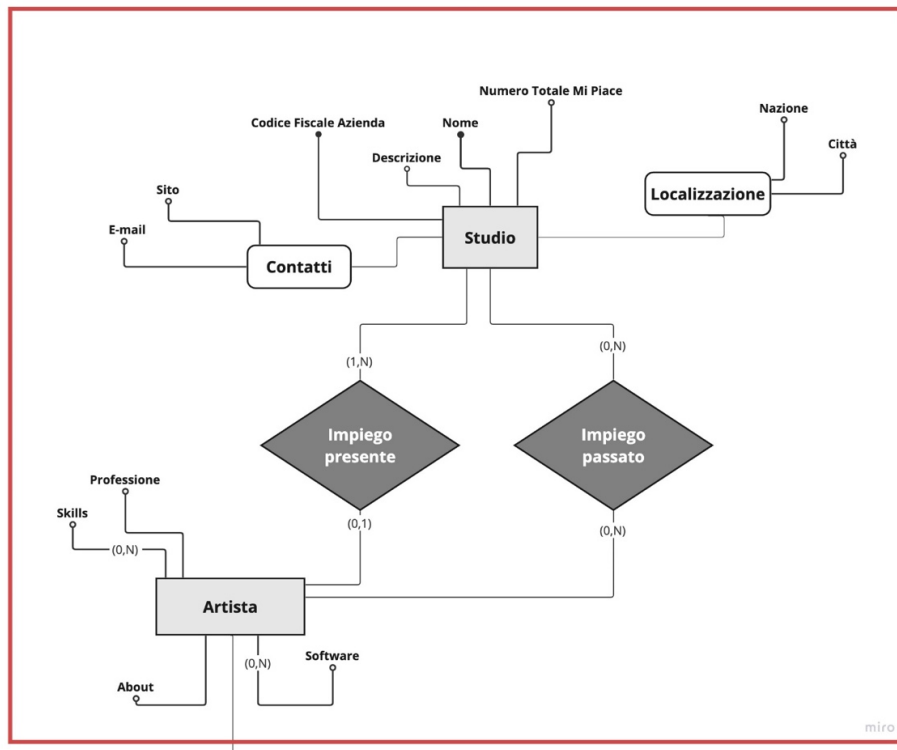


Figure 4: Prima dell'applicazione del pattern "Generalizzazione" su Impiego

### 3.2.2 Pattern "Generalizzazione"

Il pattern adottato per gestire la possibilità che una persona abbia lavorato per periodi diversi nella stessa azienda richiede un vincolo specifico. Tale vincolo impone che tutte le occorrenze della relazione "Artista Impiego" corrente devono comparire anche tra le occorrenze della relazione "Artista Impiego".

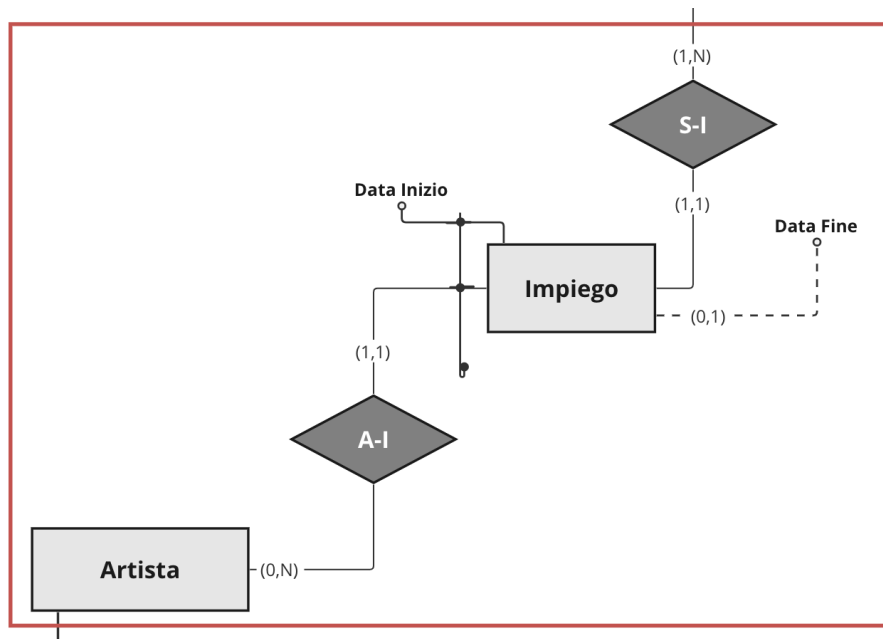


Figure 5: Dopo l'applicazione del pattern "Generalizzazione" su Impiego

### 3.3 Dizionario dei dati

Entità	Descrizione	Attributi	Identificatore
Utente	Persona registrata sulla piattaforma	Nome utente, Nome, Cognome, Età, Data di nascita, Email, Password, Città, Paese	Nome utente, Email
Artista	Individui che esprimono la loro creatività attraverso varie forme d'arte come pittura, scultura e fotografia. Gli artisti utilizzano la loro immaginazione e competenze tecniche per creare opere che evocano emozioni, trasmettono idee o rappresentano il mondo intorno a loro.	Professione, Competenze, Descrizione, Software	

<b>Entità</b>	<b>Descrizione</b>	<b>Attributi</b>	<b>Identificatore</b>
Recensione Multimediale	Recensione pubblicata sulla piattaforma da un utente artista sotto forma di post multimediale	Numero recensione, Descrizione, Titolo, Mi piace, Visualizzazioni, Tipo, Mezzo	Numero recensione, Artista
Opera d'arte	File multimediale associato alla recensione pubblicata dall'utente artista sulla piattaforma	Numero opera d'arte, Didascalia, Dati binari, Risoluzione verticale, Risoluzione orizzontale, Tipo MIME, Profilo colore	Numero opera d'arte, Recensione multimediale
Studio	Spazi dedicati a gruppi di artisti che lavorano insieme su progetti comuni. Gli studi consentono agli artisti di condividere il loro lavoro, collaborare, reclutare nuovi membri e promuovere le loro attività creative.	Descrizione, Nome, Numero di telefono, Sito web, Email, Paese, Città	Nome
Commento	Feedback pubblicato dagli utenti della piattaforma riguardo alle recensioni principali (multimediali)	Testo, Mi piace, Data di pubblicazione, Numero commento	Data di pubblicazione, Recensione multimediale
Occupazione	Coinvolgimento di un artista in uno studio	Data di inizio	Data di inizio, Artista
Occupazione attuale	Coinvolgimento attuale di un artista in uno studio		
Occupazione passata	Coinvolgimento passato di un artista in uno studio	Data di fine	
Tag	Attributo che categorizza una recensione multimediale	Nome	Nome

Table 2: Entità

Relazione	Descrizione	Entità coinvolte	Attributi
Autore di	Collega il Commento alla Persona che lo ha scritto	Artista, Commento	
Risposta	Collega il Commento secondario al Commento principale	Commento	
Mi Piace Commento	Collega la Persona al Commento che ha apprezzato	Utente, Commento	
Pubblicazione	Collega l'Utente Artista alla Recensione multimediale che sta pubblicando	Artista, Recensione multimediale	Lavoro personale, Data
Mi Piace Recensione	Collega la Persona alla Recensione multimediale che ha apprezzato	Utente, Recensione multimediale	
Feedback	Collega il Commento alla Recensione multimediale a cui si riferisce	Recensione multimediale, Commento	
Composizione	Collega l'Opera d'arte alla Recensione multimediale	Recensione multimediale, Opera d'arte	
Tag Recensione	Collega un Tag a una Recensione multimediale, categorizzandola	Tag, Recensione multimediale	
Aggiunge	Collega un Artista a un nuovo Tag	Tag, Artista	
A-Icorr	Collega l'Artista all'Occupazione attuale	Artista, Occupazione attuale	
A-I	Collega l'Artista all'Occupazione	Artista, Occupazione	
S-I	Collega lo Studio all'Occupazione	Studio, Occupazione	

Table 3: Relazioni

### 3.4 Regole Aziendali

	Vincoli
1	Le opere d'arte devono avere una risoluzione minima di 400x400 e una risoluzione massima di 10000x10000.
2	Le opere d'arte devono avere una dimensione massima di 10 MB.
3	È consentito solo un livello di risposte sotto un commento (nessuna risposta nidificata).
4	Una recensione multimediale può avere un massimo di 10 opere d'arte.
5	Una persona deve avere almeno 18 anni per essere un utente sulla piattaforma.
6	Un commento può contenere al massimo 500 caratteri.
7	Gli artisti possono aggiungere un massimo di 20 tag a ogni recensione multimediale.
8	Gli artisti possono aggiungere un massimo di 3 tipi a ogni recensione multimediale.
9	Gli artisti possono aggiungere un massimo di 3 mezzi a ogni recensione multimediale.
10	Gli artisti possono pubblicare opere non personali solo se attualmente impiegati in uno studio.
11	Le opere d'arte devono essere caricate in formati di file specifici (JPG, PNG, MP4).
12	Tutte le occorrenze della relazione Occupazione attuale dell'artista devono apparire anche nella relazione Occupazione dell'artista.
13	Le descrizioni degli artisti e degli studi non possono superare i 500 caratteri.

Table 4: Business rules

	<b>Regole di derivazione</b>
<b>1</b>	Il numero totale di "Mi piace" per una recensione multimediale è uguale alla somma delle occorrenze nella relazione "Mi piace Recensione" relative a quella recensione.
<b>2</b>	Il numero totale di "Mi piace" per un commento è uguale alla somma delle occorrenze nella relazione "Mi piace Commento" relative a quel commento.
<b>3</b>	Il numero totale di "Mi piace" per uno studio è uguale alla somma dei "Mi piace" totali di tutte le recensioni multimediale non personali pubblicate dagli artisti affiliati allo studio nell'istante corrente.

Table 5: Rules deriving from database structure

## 4 Progettazione Logica

### 4.1 Analisi delle Prestazioni

<b>Concetto</b>	<b>Tipo</b>	<b>Volume stimato</b>
Artista	E	2.000.000
Utente	E	3.000.000
S-I	R	250.000
Studio	E	100.000
Impiego	E	500.000
Impiego Corrente	E	200.000
Impiego Passato	E	300.000
A-Icorr	R	200.000
A-I	R	500.000
Commento	E	10.000.000
Autore di	R	10.000.000
Risposta	R	3.000.000
Feedback	R	10.000.000
Mi Piace Recensione	R	50.000.000
Pubblicazione	R	10.000.000
Aggiunge	R	10.000



Concetto	Tipo	Volume stimato
Tag	E	10.000
Tag Recensione	R	20.000.000
Recensione Multimediale	E	10.000.000
Composizione	R	30.000.000
Artwork	E	30.000.000
Mi Piace Commento	R	20.000.000

Table 6: Tavola dei Volumi

Abbiamo valutato il volume delle occorrenze nel database di Artstation attraverso stime e supposizioni. Le nostre considerazioni si basano sulla popolarità tra gli artisti di questa piattaforma. Abbiamo preso in considerazione diversi fattori per valutare l'entità del traffico e delle attività nel database, al fine di garantire un'adeguata gestione delle risorse e delle prestazioni del sistema.

- **Gli artisti e gli utenti:** Abbiamo effettuato stime ipotizzando che il numero totale di utenti su Artstation sia di 3 milioni, di cui 2 milioni sono artisti. Queste valutazioni riflettono la vasta portata prevista della piattaforma Artstation e il suo specifico target di artisti.
- **Commenti e risposte:** sono state stimate elevate quantità di commenti e risposte nel database di Artstation, con un numero stimato di 10 milioni di commenti e 3 milioni di risposte. Questi volumi riflettono la natura interattiva della piattaforma, in cui gli artisti possono lasciare commenti sulle opere d'arte e partecipare a discussioni.
- **Recensioni multimediali:** Nel nostro database, le recensioni multimediali sono le entità più onerose e rappresentano l'obiettivo principale della piattaforma Artstation. Sono state stimate un elevato numero di occorrenze di recensioni multimediali, approssimativamente 10 milioni. Queste stime sono state effettuate considerando che la pubblicazione delle recensioni multimediali costituisce il fulcro dell'esperienza degli utenti su Artstation.
- **Artwork:** Sono state stimate circa 30 milioni di opere d'arte nel nostro database. Questa stima tiene conto del fatto che una recensione multimediale può essere composta da più di un'opera d'arte, con una media approssimativa di circa tre opere d'arte per ogni recensione.
- **Tag:** La stima è stata fatta su 10 mila tag unici, che vanno a identificare 20 milioni di opere tramite la relazione "Tag Recensione". Questa stima tiene conto del fatto che una recensione multimediale può essere identificata da più di un tag, con una media approssimativa di due tag per ogni recensione.

Operazione	Tipo	Frequenza
Operazione 1: pubblicazione di una recensione multimediale.	I	<i>20.000 volte al giorno</i>
Operazione 2: mostrare tutte le opere personali di un determinato utente.	I	<i>100.000 volte al giorno</i>
Operazione 3: mostrare tutte le opere associate ad artisti afferenti ad un determinato studio.	I	<i>50.000 volte al giorno</i>
Operazione 4: Mostra il primo artwork relativo a tutte le recensioni multimediali che contengono una determinata parola nei campi "Descrizione", "Titolo" e "Tag" e nel campo "Didascalia" degli artwork relativi alle recensioni.	I	<i>150.000 volte a giorno</i>
Operazione 5: Acquisizione degli indirizzi e-mail degli utenti per l'aggiornamento della newsletter.	B	<i>1 volta a settimana</i>
Operazione 6: Mostrare l'artista di tendenza (artista le cui recensioni hanno ricevuto il maggior numero di "mi piace") del mese.	B	<i>1 volta al mese</i>

Table 7: Tavola delle Operazioni

### Ridondanze

- **Ridondanza 1:** numero mi piace (RECENSIONE MULTIMEDIALE). Il numero di "mi piace" relativi ad una determinata recensione multimediale si ottiene contando il numero di occorrenze dell'associazione MI PIACE RECENSIONE a cui partecipa RECENSIONE MULTIMEDIALE.

**TIPO:** Attributo derivabile da conteggio di occorrenze.

- **Ridondanza 2:** numero mi piace (COMMENTO). Il numero di "mi piace" relativi ad un determinato commento si ottiene contando il numero di occorrenze dell'associazione MI PIACE COMMENTO a cui partecipa COMMENTO.

**TIPO:** Attributo derivabile da conteggio di occorrenze.

- **Ridondanza 3:** numero totale mi piace (STUDIO). Il numero totale di "mi piace" associati ad uno specifico studio si determina contando il numero di occorrenze in cui l'associazione MI PIACE RECENSIONE si verifica tra

tutte le RECENSIONI MULTIMEDIALI che partecipano all'associazione PUBBLICAZIONE di tutti gli artisti impiegati in tale studio.

## 4.2 Eliminazione delle generalizzazioni

### 4.2.1 Generalizzazione "Utente"

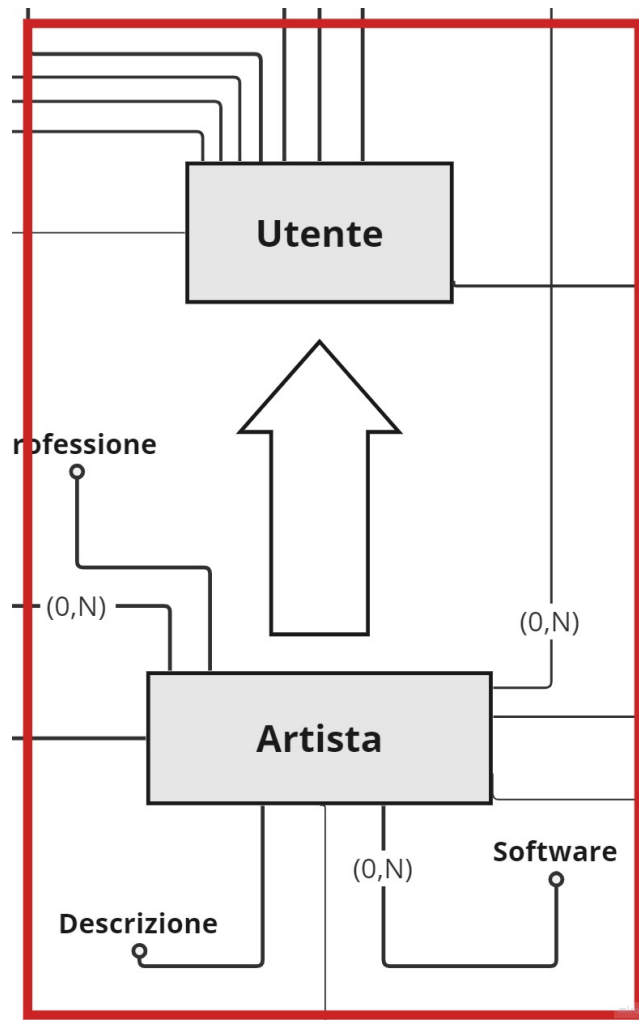


Figure 6: Prima dell'eliminazione della generalizzazione "Utente"

L'eliminazione della generalizzazione utente viene attuata mediante l'impiego della strategia di sostituzione delle generalizzazioni con associazioni. Questa scelta di strategia deriva dalla necessità di differenziare in modo esplicito Utente da Artista, in quanto vi sono operazioni che richiedono tale distinzione. Pertanto,

anziché considerare un'entità generale che rappresenta sia l'entità figlie che quella genitore, abbiamo utilizzato un'associazione per collegare le entità tra di loro.

Ciò comporta diversi vantaggi. Innanzitutto, grazie all'uso di associazioni distinte, si può gestire in modo più accurato la partecipazione delle entità a specifiche operazioni o relazioni nel database.

Un altro beneficio significativo è la riduzione dei valori nulli nel database, infatti mediante l'utilizzo delle associazioni specifiche, si può definire in modo più preciso quali attributi sono applicabili a quali entità, evitando così la presenza di valori nulli.

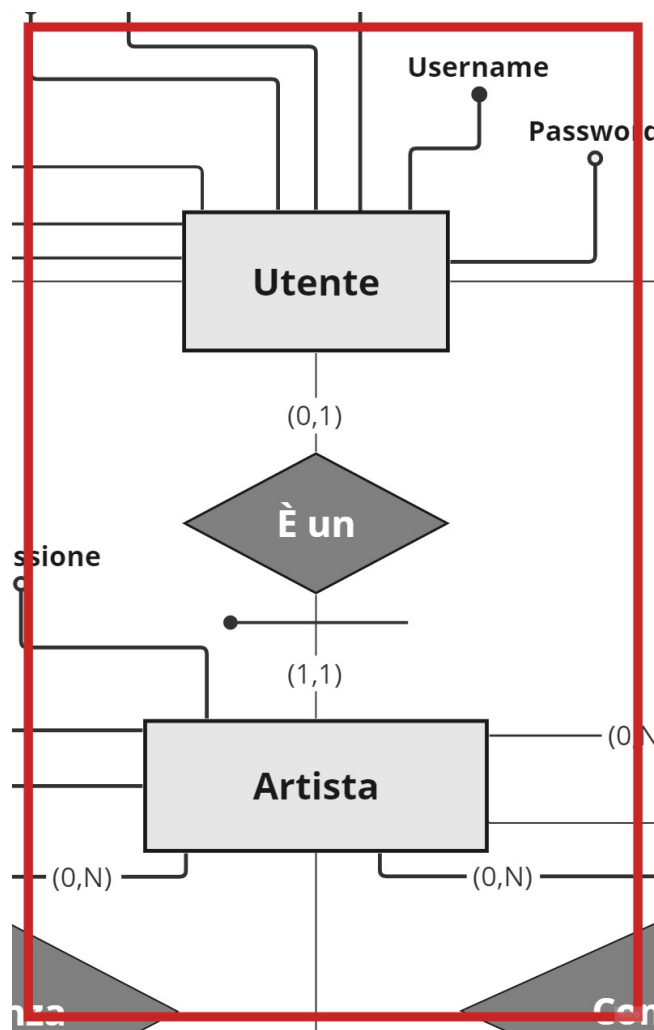


Figure 7: Dopo l'eliminazione della generalizzazione "Utente"

#### 4.2.2 Generalizzazione "Impiego"

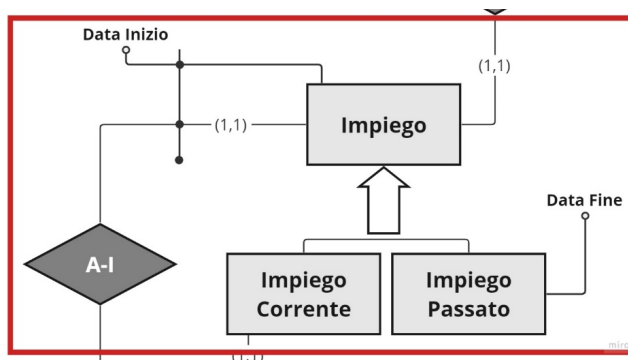


Figure 8: Prima dell'eliminazione della generalizzazione "Impiego"

Per eliminare la generalizzazione "Impiego", abbiamo optato per la strategia di accorpamento delle entità figlie nel genitore. Questa scelta è stata motivata dal fatto che le operazioni non richiedono una distinzione specifica tra l'entità padre e le entità figlie. Sebbene tale strategia abbia comportato la presenza di valori nulli nell'attributo "Data Fine" per tutti gli impieghi correnti degli artisti, abbiamo ritenuto che fosse la soluzione migliore in termini di semplicità ed efficacia per implementare uno storico degli impieghi in modo conciso. Nonostante ciò, il numero di valori nulli generati è limitato e controllato.

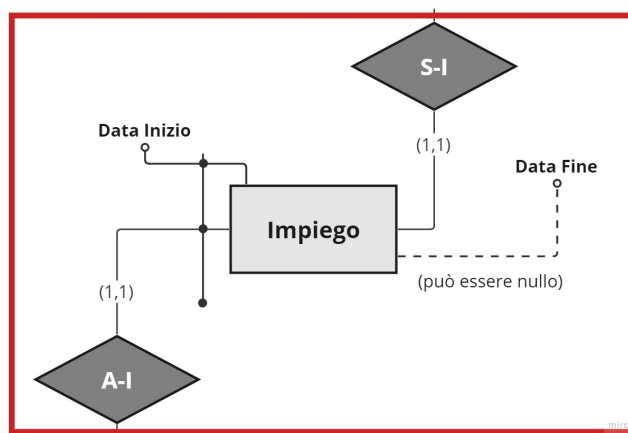


Figure 9: Dopo l'eliminazione della generalizzazione "Impiego"

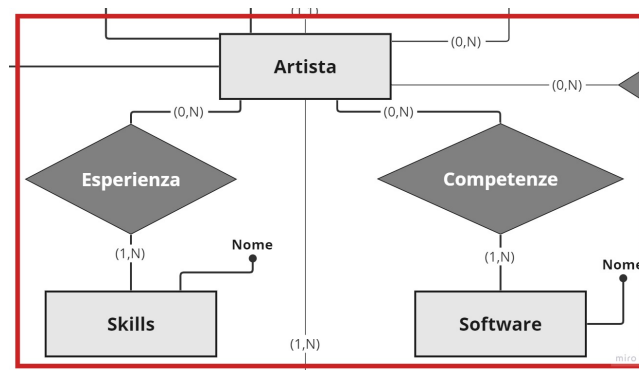


Figure 10: Eliminazione attributi multivalore su "Software" e "Skills"

### 4.3 Partizionamento/Accorpamento delle Entità e Associazioni

Abbiamo dedicato un'attenta valutazione alla questione di partizionare o accorpare entità e associazioni nel nostro database. Attraverso un'analisi approfondita delle operazioni coinvolte, siamo giunti alla conclusione che tali partizionamenti o accorpamenti non sono necessari. Durante questa valutazione, non abbiamo identificato alcun accesso parziale in cui solo determinati attributi delle entità venivano richiesti. Questo fatto ha suggerito che la suddivisione o la combinazione delle entità non fosse richiesta, poiché tutte le informazioni associate alle entità erano richiamate nelle operazioni pertinenti.

L'unico partizionamento che andremo ad effettuare all'interno del nostro database sono l'eliminazione degli attributi multivalore. L'entità **Artista** presenta due attributi multivalore "Software" e "Skills" con entrambi cardinalità (0,N) che sono state ristrutturate nel seguente modo:

L'entità **Recensione Multimediale**, invece, presenta due attributi multivalore "Tipologia" e "Medium" con entrambi cardinalità (1,N) che sono stati ristrutturati nel seguente modo:

### 4.4 Scelta degli identificatori principali

Nel contesto del nostro schema concettuale, abbiamo dedicato un'attenta valutazione per selezionare gli identificatori principali per le entità **UTENTE**, **STUDIO** e **COMMENTO**. Di seguito, forniremo una breve spiegazione delle scelte effettuate e delle motivazioni che le hanno guidate.

Per l'entità **UTENTE**, abbiamo scelto lo "username" come identificatore principale. Questa scelta è stata basata sul fatto che l'username viene ampiamente utilizzato in numerose operazioni e funzionalità chiave del sistema. L'username permette un accesso efficiente alle occorrenze degli utenti e viene spesso richiesto

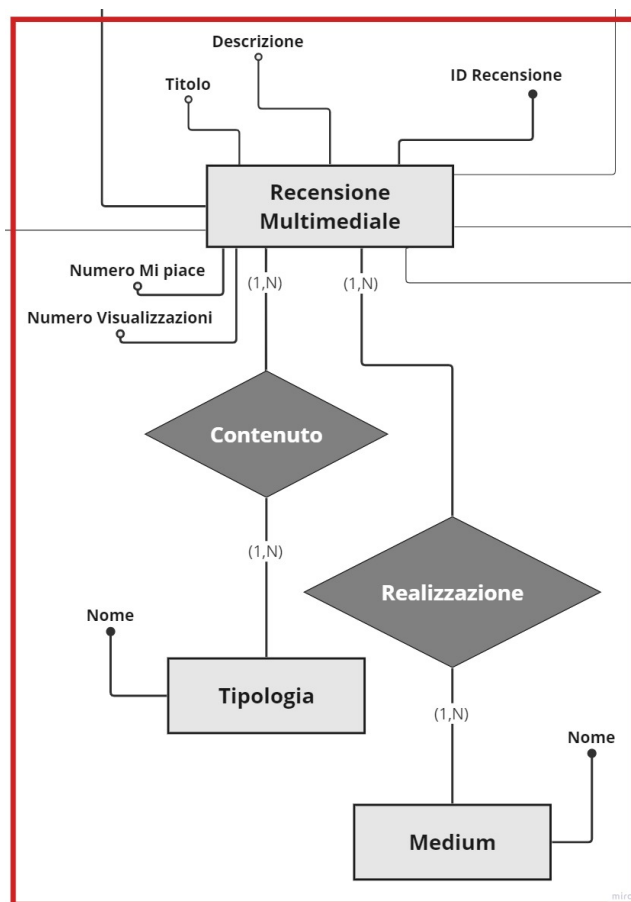


Figure 11: Eliminazione attributi multivalore su "Tipologia" e "Medium"

per l'autenticazione e l'accesso ai profili utente. Inoltre, sebbene l'attributo dell'e-mail sia rilevante, abbiamo considerato che potrebbe subire modifiche nel tempo e non viene richiesto con la stessa frequenza dell'username. È importante sottolineare che, sebbene l'username possa subire modifiche, essendo strettamente legato all'identità dell'artista e al proprio portfolio, è probabile che venga cambiato con meno frequenza rispetto all'e-mail.

Per l'entità STUDIO, abbiamo scelto il "**Codice Fiscale Azienda**" come identificatore principale. Questa decisione è stata motivata dalla sua unicità e stabilità nel tempo. Il "Codice Fiscale Azienda" rappresenta un identificatore univoco associato a ciascuno studio nel nostro schema concettuale. Inoltre, offre un collegamento significativo e pertinente all'azienda di appartenenza dello studio.

Per l'entità COMMENTO, abbiamo scelto di utilizzare un **identificatore univoco "ID"** come chiave primaria. Questa scelta ci consente di garantire un'identificazione univoca per ciascun commento nel sistema. Utilizzando un ID come chiave primaria per i commenti, siamo in grado di gestire le relazioni con altre entità in modo più efficiente, evitando la creazione di catene complesse di chiavi esterne.

Per le altre entità presenti nel database, che sono di dimensioni più ridotte e non sono al centro del focus principale, non sono state effettuate valutazioni approfondite. Ciò è dovuto al fatto che queste entità **non presentano** identificatori nulli e **non generano** complicazioni in termini di relazioni e dipendenze.

## 4.5 Schema finale ristrutturato

## 4.6 Schema logico

UTENTE (Username, E-mail, Password, Nome, Cognome, Data di nascita, Città, Nazione)

ARTISTA (Utente, Professione, Descrizione)

ESPERIENZA (Nome skill, Artista)

SKILLS (Nome)

COMPETENZE (Nome competenza, Artista)

SOFTWARE (Nome)

TAG (Nome, Artista)

TAG RECENSIONE (Tag, Recensione multimediale)

COMMENTO (ID, Numero mi piace, Testo, Data di pubblicazione, Artista, Recensione multimediale, Risposta\*)

MI PIACE RECENSIONE (Utente, Recensione multimediale)



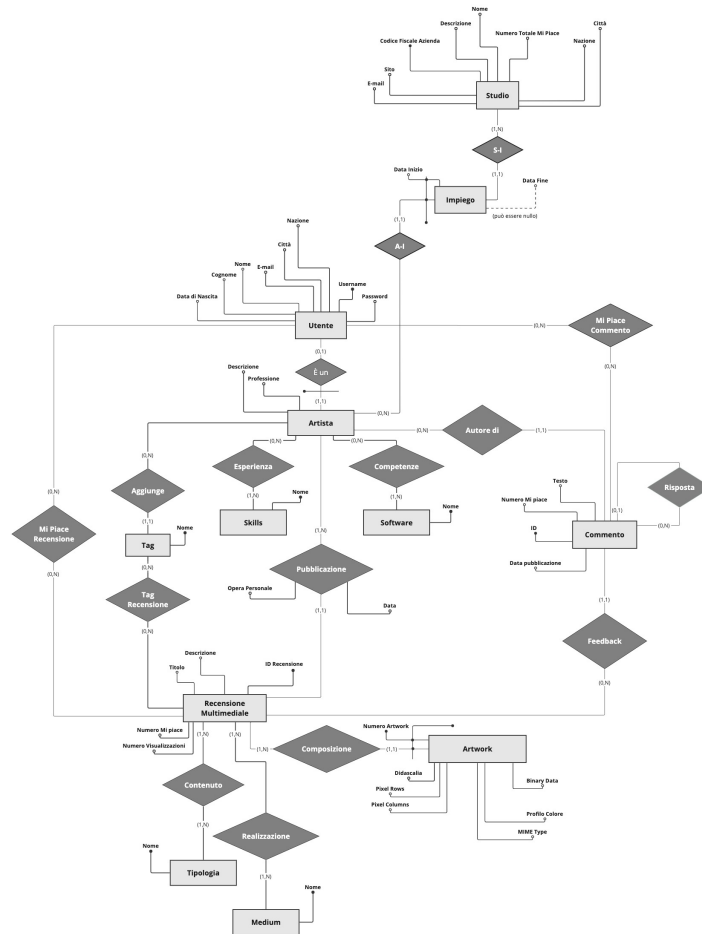


Figure 12: Schema logico ristrutturato

MI PIACE COMMENTO (Utente, Commento)

RECENSIONE MULTIMEDIALE (ID, Descrizione, Titolo, Numero mi piace, Numero Visualizzazioni, Artista, Opera Personale, Data)

ARTWORK (Recensione Multimediale, Numero Artwork, Didascalia, Binary data, Risoluzione orizzontale, Risoluzione verticale, MIME Type, Profilo colore)

CONTENUTO (Nome tipologia, Recensione multimediale)

TIPOLOGIA (Nome)

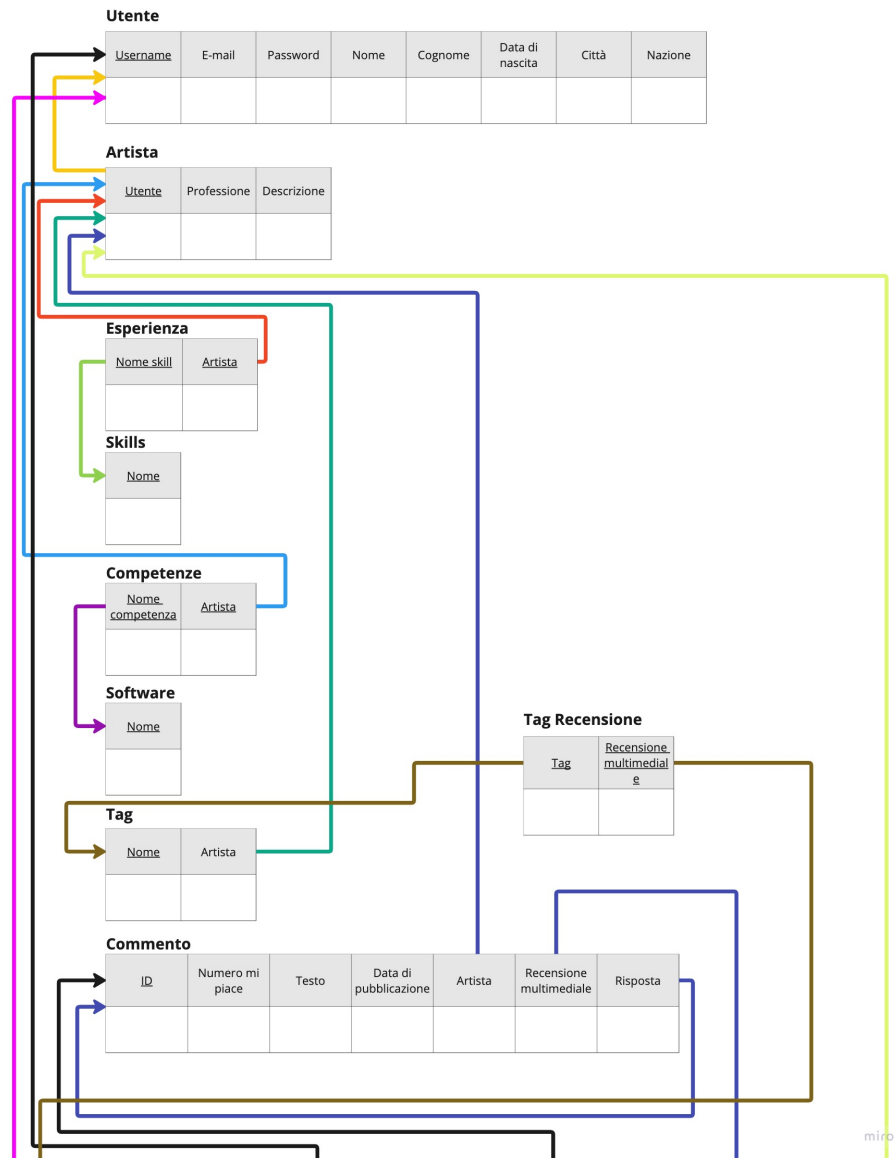
REALIZZAZIONE (Nome medium, Recensione multimediale)

MEDIUM (Nome)

IMPIEGO (Data inizio, Data fine\*, Artista, Studio)

STUDIO (Codice Fiscale Azienda, Nome, Descrizione, Numero Mi piace, E-mail, Sito, Nazione, Città)

## 4.7 Documentazione dello schema logico



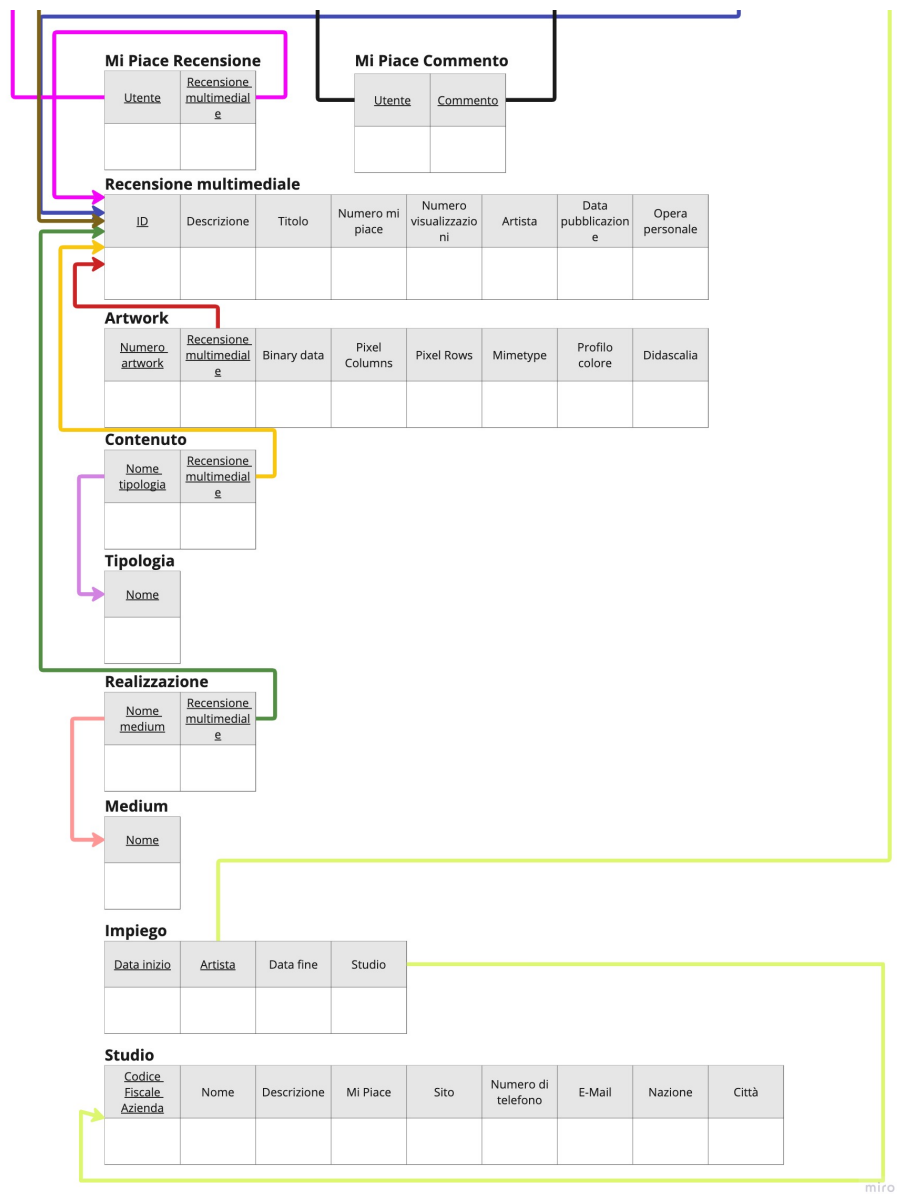


Figure 13: Documentazione schema logico

Nota: Dato il passaggio dallo schema concettuale allo schema logico dobbiamo inserire ulteriori vincoli:

(RV14) Solo gli artisti utenti della piattaforma possono lasciare commenti alle recensioni multimediali.

(RV15) Ogni occorrenza della relazione A-Icorr deve essere anche presente nella relazione A-I.

## 5 Normalizzazione

### 5.1 Verifica prima forma normale (1FN)

Per avere la prima forma normale (1FN) dobbiamo rispettare vari requisiti, i quali richiedono che il dominio degli attributi deve comprendere soltanto **valori indivisibili** (atomici) e che il valore degli attributi sia un **valore singolo** del dominio di quel rispettivo attributo. Ogni relazione del nostro database rispetta la (1FN) in quanto ha requisiti concordi con quella che è la definizione di relazione nel modello relazionale.

### 5.2 Verifica seconda forma normale (2FN)

Per il raggiungimento della seconda forma normale (2FN) è necessario che tutte le relazioni siano in (1FN) e che gli attributi non-chiave di una determinata tabella **non dipendano parzialmente** dalla chiave della tabella stessa.

Questa forma normale è quindi **verificata** a priori in quelle relazioni dove già è verificata la 1FN e la chiave è costituita da un solo attributo. Nel nostro caso, la maggior parte delle relazioni nel database presenta un unico attributo chiave, mentre gli altri attributi sono considerati non chiave, ad eccezione delle relazioni: MI PIACE RECENSIONE, MI PIACE COMMENTO, CONTENUTO, REALIZZAZIONE, IMPIEGO e ARTWORK.

Analizzando le prime quattro relazioni (MI PIACE RECENSIONE, MI PIACE COMMENTO, CONTENUTO, REALIZZAZIONE), notiamo che hanno solo due attributi, entrambi facenti parte della chiave, e quindi soddisfano già la seconda forma normale (2FN).

Nel caso della relazione IMPIEGO, dopo aver analizzato le dipendenze funzionali, possiamo osservare che tutti gli attributi non chiave dipendono esclusivamente dalla chiave primaria. Pertanto, la relazione IMPIEGO soddisfa la condizione di seconda forma normale (2FN).

Invece, per quanto riguarda la relazione ARTWORK, dopo aver esaminato le dipendenze funzionali, è evidente che tutti gli attributi non chiave dipendono solamente dalla chiave primaria. Di conseguenza, anche la relazione ARTWORK soddisfa la condizione di seconda forma normale (2FN).

Possiamo quindi confermare che tutte le relazioni rispettano i requisiti per la 2FN.

### 5.3 Verifica terza forma normale (3FN)

Per garantire il raggiungimento della terza forma normale (3FN) nel nostro database, è necessario verificare che **non** ci siano **dipendenze funzionali transitivamente** dalla chiave nelle relazioni. Quindi tutti gli attributi non-chiave devono dipendere direttamente dalla chiave.

Tale condizione è soddisfatta nelle relazioni MI PIACE RECENSIONE, MI PIACE COMMENTO, CONTENUTO, REALIZZAZIONE in quanto sono binarie poiché derivanti da associazioni.

Analizziamo di seguito le restanti relazioni:

Innanzitutto, la relazione UTENTE rispetta la 3FN in quanto gli attributi non chiave come E-mail, Password, Nome, Cognome, Data di nascita, Città e Nazione dipendono unicamente dalla chiave primaria USERNAME. Non ci sono dipendenze transitive tra gli attributi non chiave, garantendo quindi una buona organizzazione dei dati.

La relazione ARTISTA è anch'essa in terza forma normale. Gli attributi non chiave come Professione e Descrizione dipendono esclusivamente dalla chiave primaria UTENTE. Non sono presenti dipendenze transitive tra gli attributi non chiave.

Le relazioni SKILLS, SOFTWARE, MEDIUM, TIPOLOGIA sono in terza forma normale avendo un unico attributo chiave.

La relazione TAG è anch'essa in terza forma normale. Gli attributi non chiave come Artista dipendono unicamente dalla chiave primaria NOME. Non ci sono dipendenze transitive tra gli attributi non chiave.

La relazione ARTWORK è anch'essa in terza forma normale. Gli attributi non chiave come Didascalia, Binary data, Risoluzione orizzontale, Risoluzione verticale, MIME Type, Profilo colore e Dimensione dipendono esclusivamente dalla chiave primaria composta da RECENSIONE MULTIMEDIALE e NUMERO ARTWORK. Non ci sono dipendenze transitive tra gli attributi non chiave.

La relazione STUDIO rispetta la terza forma normale. Gli attributi non chiave come Nome, Descrizione, Numero Mi piace, E-mail, Sito, Nazione e Città dipendono unicamente dalla chiave primaria CODICE FISCALE AZIENDA. Non sono presenti dipendenze transitive tra gli attributi non chiave.

Infine, la relazione IMPIEGO soddisfa la 3FN. Gli attributi non chiave come Data inizio, Data fine e Studio dipendono esclusivamente dalla chiave primaria composta da Data inizio, Artista e Studio. Non sono presenti dipendenze transitive tra gli attributi non chiave.

Complessivamente, l'analisi delle dipendenze funzionali nelle relazioni indica che esse sono strutturate in modo da soddisfare la terza forma normale, non presentando dipendenze funzionali transitive e garantendo che gli attributi non chiave dipendano solo dalla chiave primaria.

## 5.4 Verifica forma normale di Boyce-Codd

Una relazione  $r$  è in forma normale di **Boyce-Codd** se per ogni dipendenza funzionale (non banale)  $X \rightarrow A$  definita su di essa,  $X$  contiene una chiave  $K$  di  $r$ , cioè  $X$  è superchiave per  $r$ .

Possiamo determinare se il nostro schema soddisfa la forma normale di Boyce-Codd (BCNF) verificando che, nelle dipendenze funzionali analizzate in precedenza,  $X$  sia una superchiave.

Dopo un'attenta analisi, possiamo affermare che lo schema logico è in BCNF, poiché non sono state trovate dipendenze funzionali non banali che violino le condizioni di BCNF. Ciò indica che il database è ben strutturato e rispetta le proprietà di integrità dei dati nella forma normale di BCNF.

## 6 Query SQL

### 6.1 Query con operatore di aggregazione e join: Classifica impiegati dell'anno

La query esegue un'operazione di join tra diverse tabelle per ottenere una classifica degli impiegati correnti del mese in vari studi. La classifica viene calcolata utilizzando la media dei "Mi piace" ricevuti sulle recensioni multimediali pubblicate sotto ciascuno studio, la media dei "Mi piace" ricevuti dai commenti su tali recensioni e la media delle visualizzazioni delle recensioni stesse. L'obiettivo è identificare gli impiegati che hanno ottenuto il maggior coinvolgimento e apprezzamento per il loro lavoro durante il mese considerato.

```
SELECT
  rm.artista,
  u.nome,
  u.cognome,
  u.data_di_nascita AS "data di nascita",
  s.nome "studio",
  Round(
    Avg(rm.numero_mi_piace),
    2
  ) AS avg_mipiacerecensione,
  Round(
    Avg(c.numero_mi_piace),
    2
  ) AS avg_mipiacecommenti,
  Round(
    Avg(numero_visualizzazioni),
    2
  ) AS avg_visualizzazione
FROM
  recensione_multimediale AS rm
  JOIN commento c ON rm.id_recensione_multimediale =
    ⇐ c.recensione_multimediale
  AND risposta IS NULL
  JOIN utente AS u ON u.username = rm.artista
```



```

    JOIN impiego AS i ON i.artista = u.username
    JOIN studio AS s ON s.codice_fiscale_azienza = i.studio
WHERE
    rm.opera_personale = false 56 AND i.data_fine IS NULL
    AND rm.data_pubblicazione <= Now()
    AND rm.data_pubblicazione >= Now() - interval '1 year'
    AND c.data_di_pubblicazione <= now()
    AND c.data_di_pubblicazione >= now() - interval '1 year'
GROUP BY
    rm.artista,
    s.nome,
    u.nome,
    u.cognome,
    u.data_di_nascita
ORDER BY
    avg_visualizzazione recensione DESC,
    avg_mi piace recensione DESC,
    avg_mi piace commenti DESC;

```

## 6.2 Query nidificata complessa: Ricerca artisti versatili

La query ha come scopo quello di andare a selezionare, se presenti, tutti gli artisti (insieme alle loro specializzazioni), i quali sono stati impiegati almeno una volta in tutti gli studio della piattaforma.

```

SELECT
    a.utente,
    a.professione AS specializzazione
FROM
    artista a
WHERE
    NOT EXISTS (
        SELECT
            *
        FROM
            studio s
        WHERE
            NOT EXISTS (
                SELECT
                    *
                FROM
                    impiego i
                WHERE
                    i.artista = a.utente
                    AND i.studio = s.codice_fiscale_azienza
            )
    )

```

```
)
);
```

### 6.3 Query insiemistica: Arti tradizionali

Questa query restituisce l'elenco degli artisti che dimostrano un forte interesse nelle arti tradizionali. Vengono inclusi gli artisti che hanno interagito mettendo "mi piace" sia a recensioni che a commenti relativi a contenuti multimediali tradizionali, nonché quelli che hanno pubblicato recensioni multimediali tradizionali.

```
utente,
real.nome_medium
FROM
mi_piace_recensione AS mpr
JOIN recensione_multimediale AS r ON
    ⇨ mpr.recensione_multimediale = r.id_recensione_multimediale
JOIN realizzazione AS real ON real.recensione_multimediale =
    ⇨ r.id_recensione_multimediale
WHERE
real.nome_medium LIKE 'Traditional%'
INTERSECT
SELECT
utente,
real.nome_medium
FROM
mi_piace_commento AS mpc
JOIN commento AS c ON c.id_commento = mpc.commento
JOIN recensione_multimediale AS rm ON
    ⇨ rm.id_recensione_multimediale = c.recensione_multimediale
JOIN realizzazione AS real ON real.recensione_multimediale =
    ⇨ rm.id_recensione_multimediale
WHERE
real.nome_medium LIKE 'Traditional%'
INTERSECT
SELECT
r.artista,
real.nome_medium
FROM
recensione_multimediale AS r
JOIN realizzazione AS real ON r.id_recensione_multimediale =
    ⇨ real.recensione_multimediale
WHERE
real.nome_medium LIKE 'Traditional%'
```

## 6.4 Eventuali Altre query

### 6.4.1 Ricerca competenze specifiche

Questa query seleziona gli utenti artisti che hanno competenze specifiche in software come 'Blender' e 'Cinema 4D', escludendo coloro che hanno competenze nel campo del "Game Art".

```
SELECT
    u.username,
    u.nome,
    u.cognome
FROM
    utente u
    JOIN artista a ON u.username = a.utente
    JOIN competenza c ON a.utente = c.artista
WHERE
    c.nome_competenza IN ('Blender', 'Cinema 4D')
INTERSECT
SELECT
    u.username,
    u.nome,
    u.cognome
FROM
    utente u
    JOIN artista a ON u.username = a.utente
    JOIN esperienza e ON a.utente = e.artista
    JOIN skills s ON e.nome_skill = s.nome
WHERE
    s.nome <> 'Game Art';
```

## 7 Viste

### 7.1 Vista *recensione\_studio*

La vista "recensione\_studio" permette di identificare l'appartenenza di una specifica recensione multimediale a un particolare studio tra tutte le recensioni multimediali non personali. Inoltre, fornisce informazioni generali sulla recensione.

```
CREATE VIEW recensione_studio AS (
    SELECT
        r.id_recensione_multimediale,
        r.titolo,
        r.descrizione,
        r.artista,
        s.nome as "studio_di_publicazione",
```

```

        r.data_pubblicazione,
        r.numero_mi_piace,
        r.numero_visualizzazioni
FROM
    recensione_multimediale r
    JOIN impiego i ON i.artista = r.artista
    JOIN studio s ON s.codice_fiscale_azienza = i.studio
WHERE
    CASE WHEN i.data_fine IS NULL THEN r.data_pubblicazione >
        ⇨ i.data_inizio
    AND r.opera_personale = 'f' WHEN i.data_fine IS NOT NULL THEN
        ⇨ r.data_pubblicazione > i.data_inizio
    AND r.data_pubblicazione < i.data_fine
    AND r.opera_personale = 'f' END
)

```

### 7.1.1 Query con Vista: Impiegato più produttivo sulla piattaforma

Questa query, basata sulla vista *recensione\_studio*, restituisce l'artista più produttivo all'interno di uno studio specifico. Identifica l'impiegato che ha pubblicato il maggior numero di opere sulla piattaforma, fornendo anche il nome dello studio associato a tali pubblicazioni e il conteggio esatto delle opere pubblicate.

```

SELECT
    studio_di_pubblicazione,
    artista AS "impiegato_più_produttivo",
    COUNT(
        DISTINCT id_recensione_multimediale
    ) AS "numero_di_opere_pubblicate"
FROM
    recensione_studio
GROUP BY
    studio_di_pubblicazione,
    artista
HAVING
    COUNT(
        DISTINCT id_recensione_multimediale
    ) = (
        SELECT
            MAX(recensioni_pubblicate)
        FROM
            (
                SELECT
                    COUNT(
                        DISTINCT id_recensione_multimediale

```

```

        ) AS recensioni_pubblicate
FROM
    recensione_studio r
GROUP BY
    artista
    ) AS numero_recensioni_per_impiegato
)

```

## 7.2 Vista *stato\_studio*

La vista in questione (*stato\_studio*) mostra la lista dei vari “studio” con annesso il numero di impiegati correnti, il numero di impiegati passati, il numero totale di recensioni multimediali pubblicate da artisti afferenti a quello studio, il numero di visualizzazioni totali ed il numero di “mi piace” totali.

Per realizzare tale vista si è fatto uso di una vista di supporto (*recensioni\_studio*) la quale mostra tutti gli impiegati di ogni studio:

```

CREATE
OR REPLACE VIEW stato_studio AS
SELECT
    s.codice_fiscale_azienza,
    s.nome,
    COUNT(
        CASE WHEN i.data_fine IS NULL THEN 1 END
    ) AS "#dipendenti_presenti",
    COUNT(
        CASE WHEN i.data_fine IS NOT NULL THEN 1 END
    ) AS "#dipendenti_passati",
    s.numero_mi_piace AS "#numero_mi_piace",
    SUM(rs.numero_visualizzazioni) AS "#numero_visualizzazioni",
    COUNT() DISTINCT rs.id_recensione_multimediale
FROM
    studio AS s
    JOIN impiego AS i ON i.studio = s.codice_fiscale_azienza
    LEFT JOIN recensione_studio AS rs ON rs.codice_fiscale_studio =
        ⇨ s.codice_fiscale_azienza
HAVING
    COUNT(
        DISTINCT id_recensione_multimediale
    ) = (
        SELECT
            MAX(recensioni_pubblicate)
        FROM
            (
                SELECT
                    COUNT(

```

```

        DISTINCT id_recensione_multimediale
    ) AS recensioni_pubblicate
FROM
    recensione_studio r
GROUP BY
    artista
) AS numero_recensioni_per_impiegato
)
GROUP BY
    s.codice_fiscale_azienda,
    s.nome

```

## 8 Trigger

### 8.1 Trigger inizializzazione

#### 8.1.1 Trigger: Controllo Recensione Multimediale

Il trigger controllo\_recensione\_multimediale scatta durante la fase di popolarimento/inizializzazione del database e ha lo scopo di verificare alcune condizioni di validità della recensione. Quest'ultimo verifica il rispetto dei limiti stabiliti per le varie entità che compongono la recensione multimediale. In particolare, controlla se il numero di artwork sia compreso tra 1 e 10, se il numero di medium sia compreso tra 1 e 3, se il numero di tipologie sia compreso tra 1 e 3 ed infine se il numero di tag sia al massimo 20. Se una qualsiasi delle condizioni di validità non viene soddisfatta, viene generata un'eccezione.

```

--trigger_function
CREATE
OR REPLACE FUNCTION controllo_recensione_multimediale() RETURNS
trigger AS $$ DECLARE num_artwork INTEGER;
num_medium INTEGER;
num_tipologia INTEGER;
num_tag INTEGER;
BEGIN
SELECT
    COUNT(*) INTO num_artwork
FROM
    artwork a
WHERE
    a.recensione_multimediale = NEW.id_recensione_multimediale;
SELECT
    COUNT(*) INTO num_medium
FROM
    realizzazione r

```

```

WHERE
    r.recensione_multimediale = NEW.id_recensione_multimediale;
SELECT
    COUNT(*) INTO num_tipologia
FROM
    contenuto c
WHERE
    c.recensione_multimediale = NEW.id_recensione_multimediale;
SELECT
    COUNT(*) INTO num_tag
FROM
    tag_recensione tr
WHERE
    62 tr.recensione_multimediale = NEW.id_recensione_multimediale;
IF (num_artwork = 0) THEN RAISE EXCEPTION 'Impossibile inserire
una recensione senza artwork';
END IF;
IF (num_medium = 0) THEN RAISE EXCEPTION 'Impossibile inserire
↪ una
recensione senza specificare un medium';
END IF;
IF (num_tipologia = 0) THEN RAISE EXCEPTION 'Impossibile inserire
una recensione senza specificare una tipologia';
END IF;
IF (num_artwork > 10) THEN RAISE EXCEPTION 'Impossibile inserire
più di dieci artwork';
END IF;
IF (num_medium > 3) THEN RAISE EXCEPTION 'Impossibile inserire
↪ più
di tre medium';
END IF;
IF (num_tipologia > 3) THEN RAISE EXCEPTION 'Impossibile inserire
più di tre tipologie';
END IF;
IF (num_tag > 20) THEN RAISE EXCEPTION 'Impossibile inserire più
di 20 tag';
END IF;
RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
--trigger
CREATE CONSTRAINT TRIGGER recensione_valida
AFTER
    INSERT ON recensione_multimediale DEFERRABLE INITIALLY DEFERRED
    ↪ FOR EACH ROW EXECUTE PROCEDURE
    ↪ controllo_recensione_multimediale();

```

### 8.1.2 Trigger: Controlla Risposta

Il trigger controlla\_risposta scatta nel momento in cui viene inserito o modificato un commento. Quest'ultimo si occupa di controllare se il commento che si sta inserendo non sia una risposta ad una risposta ed in tal caso genera un'eccezione.

```
--trigger-function
CREATE
OR REPLACE FUNCTION controlla_risposta() RETURNS TRIGGER AS $$
    BEGIN IF NEW.risposta IS NOT NULL THEN IF (
        SELECT
            risposta
        FROM
            commento
        WHERE
            id_commento = NEW.risposta
    ) IS NOT NULL THEN RAISE EXCEPTION 'Non puoi inserire una
    ↳ risposta
ad una risposta id_commento=%',
NEW.id_commento;
END IF;
IF (
    SELECT
        recensione_multimediale
    FROM
        commento
    WHERE
        id_commento = NEW.risposta
    ) <> NEW.recensione_multimediale THEN RAISE EXCEPTION 'La
recensione_multimediale della risposta non corrisponde alla
recensione_multimediale del commento principale con
id_commento=%',
NEW.id_commento;
END IF;
END IF;
RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
-- trigger
CREATE TRIGGER controlla_risposta
AFTER
    INSERT
    OR
    UPDATE
```



```
ON commento FOR EACH ROW EXECUTE PROCEDURE
↳ controlla_risposta();
```

### 8.1.3 Trigger: Controllo Impiegati Studio

Il trigger controllo\_impiegati\_studio scatta nel momento in cui viene inserito uno studio. Quest'ultimo si occupa di controllare, se la cardinalità minima viene rispettata, ossia se almeno un artista ha un impiego.

```
-- trigger function
CREATE
OR REPLACE FUNCTION controllo_impiegati_studio() RETURNS TRIGGER
↳ AS $$ BEGIN IF NOT EXISTS (
  SELECT
    *
  FROM
    impiego i
  WHERE
    NEW.codice_fiscale_azienza = i.studio
) THEN RAISE EXCEPTION 'Lo studio "%", con codice fiscale "%",
deve avere almeno un impiegato',
NEW.nome,
NEW.codice_fiscale_azienza;
END IF;
RETURN NULL;
END;
$$ LANGUAGE PLPGSQL;
-- trigger
CREATE CONSTRAINT TRIGGER studio_valido
AFTER
  INSERT ON studio DEFERRABLE INITIALLY DEFERRED FOR EACH ROW
↳ EXECUTE PROCEDURE controllo_impiegati_studio();
```

### 8.1.4 Trigger: Controllo Artista

Il trigger controllo\_artista scatta nel momento in cui viene inserito un artista. Quest'ultimo si occupa di controllare, se la cardinalità minima viene rispettata, ossia se l'artista ha almeno una recensione multimediale.

```
-- trigger function
CREATE
OR REPLACE FUNCTION controllo_artista() RETURNS TRIGGER AS $$
↳ BEGIN IF NOT EXISTS (
  SELECT
    *
```

```

FROM
    recensione_multimediale r
WHERE
    NEW.utente = r.artista
) THEN RAISE EXCEPTION 'L''artista "%" deve aver pubblicato
↳ almeno
una recensione multimediale',
NEW.utente;
END IF;
RETURN NULL;
END;
$$ LANGUAGE PLPGSQL;
-- trigger
CREATE CONSTRAINT TRIGGER artista_valido
AFTER
    INSERT ON artista DEFERRABLE INITIALLY DEFERRED FOR EACH ROW
↳ EXECUTE PROCEDURE controllo_artista();

```

### 8.1.5 Trigger: Aggiungi Numero Artwork

Il trigger aggiungi\_numero\_artwork scatta prima dell'inserimento di un artwork. Quest'ultimo si occupa di inserire un numero all'artwork che si sta inserendo, in maniera progressiva, in quella determinata recensione multimediale.

```

-- trigger function
CREATE
OR REPLACE FUNCTION aggiungi_numero_artwork() RETURNS TRIGGER AS
↳ $$ BEGIN NEW.numero_artwork := (
    SELECT
        COUNT(*) + 1
    FROM
        artwork
    WHERE
        recensione_multimediale = NEW.recensione_multimediale
);
RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
-- trigger
CREATE TRIGGER aggiungi_numero_artwork BEFORE INSERT ON artwork
↳ FOR EACH ROW EXECUTE PROCEDURE aggiungi_numero_artwork();

```

## 8.2 Trigger per vincoli aziendali

### 8.2.1 Trigger: Aggiungi Mi Piace Commento

Il trigger `aggiungi_mi_piace_commento` scatta nel momento che viene inserito un “mi piace” ad un commento. Quest’ultimo si occupa di aggiornare il numero dei “mi piace” totali di quel determinato commento, aumentando il valore di 1.

```
--trigger_function
CREATE
OR REPLACE FUNCTION aggiungi_mi_piace_commento() RETURNS trigger
↪ AS $$ BEGIN
UPDATE
    commento SET numero_mi_piace = numero_mi_piace + 1
WHERE
    id_commento = NEW.commento;
RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
--trigger
CREATE TRIGGER aggiungi_commento
AFTER
    INSERT ON mi_piace_commento FOR EACH ROW EXECUTE PROCEDURE
    ↪ aggiungi_mi_piace_commento();
```

### 8.2.2 Trigger: Elimina Mi Piace Commento

Il trigger `elimina_mi_piace_commento` scatta nel momento che viene rimosso un “mi piace” ad un commento. Quest’ultimo si occupa di aggiornare il numero dei “mi piace” totali di quel determinato commento, diminuendo il valore di 1.

```
--trigger_function
CREATE
OR REPLACE FUNCTION elimina_mi_piace_commento() RETURNS trigger
↪ AS $$ BEGIN
UPDATE
    commento SET numero_mi_piace = numero_mi_piace - 1
WHERE
    id_commento = OLD.commento;
RETURN OLD;
END;
$$ LANGUAGE PLPGSQL;
--trigger
CREATE TRIGGER elimina_mi_piace_commento
AFTER
    DELETE ON mi_piace_commento FOR EACH ROW EXECUTE PROCEDURE
    ↪ elimina_mi_piace_commento();
```

### 8.2.3 Trigger: Aggiungi Mi Piace Recensione

Il trigger `aggiungi_mi_piace_recensione` scatta nel momento che viene inserito un “mi piace” ad una recensione multimediale. Quest’ultimo si occupa di aggiornare il numero dei “mi piace” totali di quella determinata recensione, aumentando il valore di 1.

```
--trigger_function
CREATE
OR REPLACE FUNCTION aggiungi_mi_piace_recensione() RETURNS
    trigger AS $$ BEGIN
UPDATE
    recensione_multimediale SET numero_mi_piace = numero_mi_piace +
        1
WHERE
    id_recensione_multimediale = NEW.recensione_multimediale;
RETURN NEW;
END;
$$ LANGUAGE PLPGSQL;
--trigger
CREATE TRIGGER aggiungi_mi_piace_recensione
AFTER
    INSERT ON mi_piace_recensione FOR EACH ROW EXECUTE PROCEDURE
        aggiungi_mi_piace_recensione();
```

### 8.2.4 Trigger: Elimina Mi Piace Recensione

Il trigger `elimina_mi_piace_recensione` scatta nel momento che viene rimosso un “mi piace” ad una recensione multimediale. Quest’ultimo si occupa di aggiornare il numero dei “mi piace” totali di quella determinata recensione, diminuendo il valore di 1.

```
--trigger_function
CREATE
OR REPLACE FUNCTION elimina_mi_piace_recensione() RETURNS trigger
    AS $$ BEGIN
UPDATE
    recensione_multimediale SET numero_mi_piace = numero_mi_piace -
        1
WHERE
    id_recensione_multimediale = OLD.recensione_multimediale;
RETURN OLD;
END;
$$ LANGUAGE PLPGSQL;
--trigger
```

```

CREATE TRIGGER elimina_mi_piace_recensione
AFTER
DELETE ON mi_piace_recensione FOR EACH ROW EXECUTE PROCEDURE
    ↪ elimina_mi_piace_recensione();

```

### 8.2.5 Trigger: Aggiorna Mi Piace Studio

Il trigger aggiorna \_numero\_mi\_piace\_studio scatta nel momento in cui viene inserito un “mi piace” ad una recensione multimediale. Quest’ultimo si occupa di aggiornare il numero dei “mi piace” totali dello studio sotto il quale è stata pubblicata tale recensione multimediale ricalcolando la somma.

```

--trigger-function
CREATE
OR REPLACE FUNCTION aggiorna_numero_mi_piace_studio() RETURNS
    ↪ TRIGGER AS $$ DECLARE studio_pubblicazione VARCHAR(30);
BEGIN
SELECT
    i.studio INTO studio_pubblicazione
FROM
    recensione_multimediale r
    JOIN impiego i ON i.artista = r.artista
    JOIN studio s ON s.codice_fiscale_azienza = i.studio
WHERE
    CASE WHEN i.data_fine IS NULL THEN r.data_pubblicazione >
        ↪ i.data_inizio
    AND r.opera_personale = 'f' WHEN i.data_fine IS NOT NULL THEN
        ↪ r.data_pubblicazione > i.data_inizio
    AND r.data_pubblicazione < i.data_fine
    AND r.opera_personale = 'f' END
    AND r.id_recensione_multimediale =
        ↪ NEW.id_recensione_multimediale;
IF studio_pubblicazione IS NOT NULL THEN
UPDATE
    studio SET numero_mi_piace = (
        SELECT
            SUM(r.numero_mi_piace)
        FROM
            recensione_multimediale r
            JOIN impiego i ON i.artista = r.artista
            JOIN studio s ON s.codice_fiscale_azienza = i.studio
        WHERE
            CASE WHEN i.data_fine IS NULL THEN r.data_pubblicazione >
                ↪ i.data_inizio

```

```

        AND r.opera_personale = 'f' WHEN i.data_fine IS NOT NULL
        ⇨ THEN r.data_pubblicazione > i.data_inizio
        AND r.data_pubblicazione < i.data_fine
        AND r.opera_personale = 'f' END
        AND s.codice_fiscale_azienza = studio_pubblicazione
    GROUP BY
        s.codice_fiscale_azienza
)
WHERE
    codice_fiscale_azienza = studio_pubblicazione;
END IF;
RETURN NULL;
END;
$$ LANGUAGE PLPGSQL;
--trigger
CREATE CONSTRAINT TRIGGER aggiorna_numero_mi_piace_studio
AFTER
UPDATE
OF numero_mi_piace ON recensione_multimediale DEFERRABLE
⇨ INITIALLY DEFERRED FOR EACH ROW EXECUTE PROCEDURE
⇨ aggiorna_numero_mi_piace_studio();

```

### 8.2.6 Trigger: Controllo Opera Personale

Il trigger controllo\_opera\_personale scatta nel momento in cui viene inserito o modificato l'attributo opera\_personale all'interno di una recensione multimediale. Quest'ultimo si occupa di controllare, nel caso in cui venga impostata tale recensione come non personale, se quell'artista nel momento in cui ha pubblicato la recensione aveva un impiego presso uno studio.

```

-- trigger function
CREATE
OR REPLACE FUNCTION controllo_opera_personale() RETURNS TRIGGER
⇨ AS $$ BEGIN IF NEW.opera_personale = 'f' THEN IF NOT EXISTS (
    SELECT
        *
    FROM
        studio s
        JOIN impiego i ON s.codice_fiscale_azienza = i.studio
    WHERE
        i.artista = NEW.artista
        AND (
            (i.data_fine IS NULL)
            AND NEW.data_pubblicazione > i.data_inizio
            OR (

```

```

        i.data_fine IS NOT NULL
        AND NEW.data_pubblicazione > i.data_inizio
    )
)
    AND NEW.data_pubblicazione < i.data_fine
) THEN RAISE EXCEPTION 'Impossibile inserire l''opera con ID %
come non personale se l''artista non è associato a uno studio
nella data di pubblicazione',
NEW.id_recensione_multimediale;
END IF;
END IF;
RETURN NULL;
END;
$$ LANGUAGE PLPGSQL;
CREATE CONSTRAINT TRIGGER controllo_opera_personale
AFTER
    INSERT
    OR
    UPDATE
OF opera_personale ON recensione_multimediale DEFERRABLE
    ↪ INITIALLY DEFERRED FOR EACH ROW EXECUTE PROCEDURE
    ↪ controllo_opera_personale();

```

---