# Laboratorio di Basi di Dati: *Progetto “Collectors”*

**Gruppo di lavoro**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Matricola** | **Nome** | **Cognome** | **Contributo al progetto** |
| 278661 | Luca Francesco | Macera |  |
| 279660 | Michael | Piccirilli |  |
|  | Calogero | Carlino |  |

**Data di consegna del progetto**: gg/mm/aaaa

## Analisi dei requisiti

* E’ possibile riportare in questa sezione i **requisiti** copiati dal documento di specifica, oppure semplicemente riassumerne gli aspetti più importanti. Vanno quindi (eventualmente) discusse tutte le scelte progettuali relative al dominio, le ambiguità e il modo in cui sono state risolte.

Il caso d’uso rappresenta una situazione in cui si vuole tenere traccia delle **collezioni** relative ad un **collezionista**. Tali collezioni rappresentano un gruppo di **dischi** , i quali presentano ognuno una tracklist formata di **tracce**  e delle **immagini** e ulteriori informazioni. Inoltre, ogni traccia è composta da un **autore**, il quale può essere un **artista singolo** o una **band**.

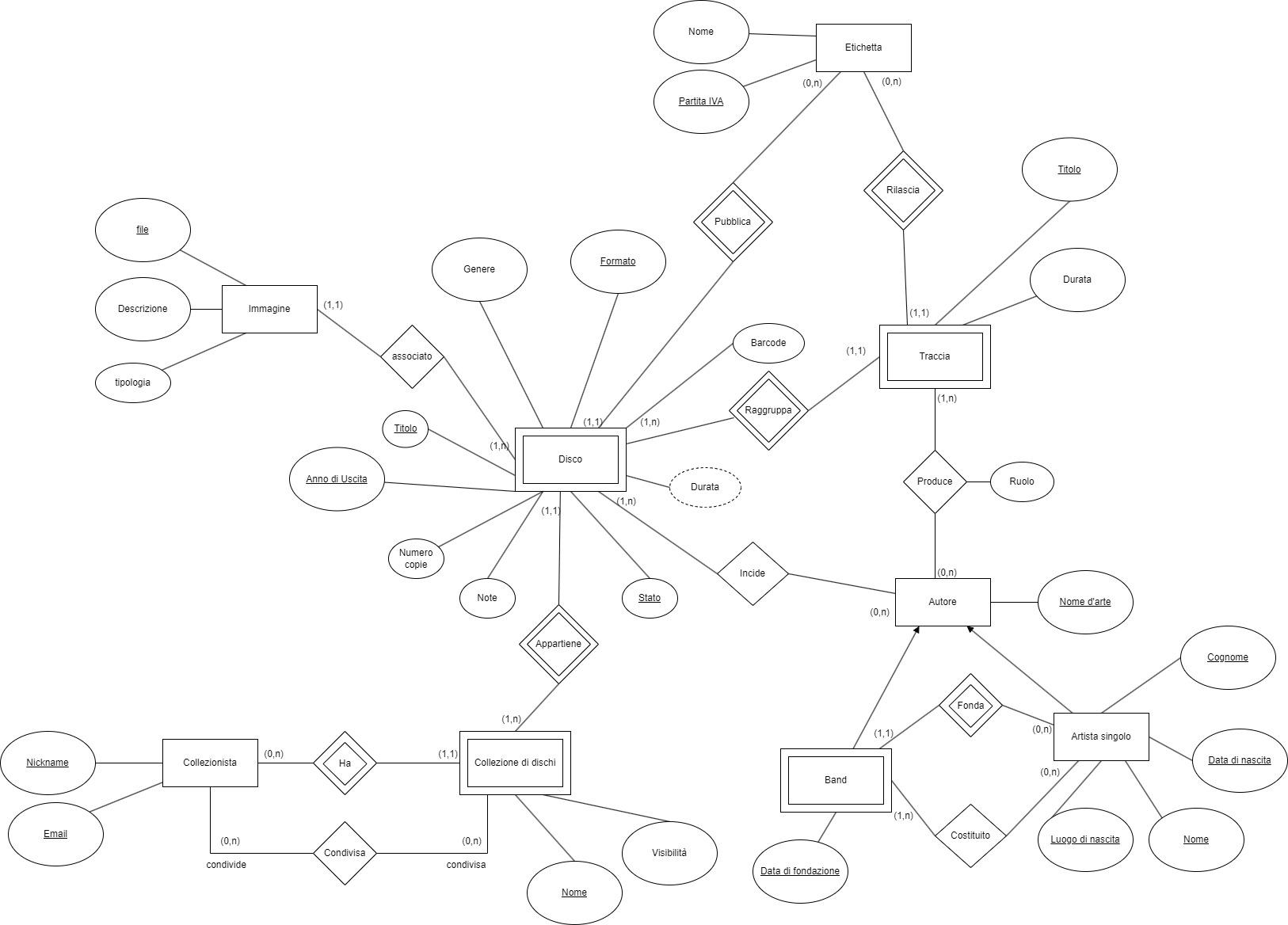
* E’ possibile infine inserire qui un glossario che riporta tutti gli oggetti di dominio individuati, con la loro semantica, i loro eventuali sinonimi e le loro proprietà.

**Dominio:**

* **Collezionista**: persona identificata univocamente da un nickname e da un indirizzo email che crea delle collezioni le quali possono essere anche condivise con altri collezionisti.
* **Collezione:** insieme di dischi che presenta nomi distinti e che può, o no, essere condivisa con altri collezionisti (infatti altri collezionisti potranno vedere la collezione ma non apportare modifiche a quest’ultima)
* **Disco**: rappresenta in tutto e per tutto il concetto di disco, il quale presenta un autore, un titolo, un anno di uscita, un’etichetta, dei generi, lo stato di conservazione, il formato, il barcode (il quale potrebbe essere anche assente) e il numero di copie presenti nella collezione di cui fa parte
* **Immagine:** relativa ad un disco, può essere la copertina, il retro o ulteriori tipologia.
* **Etichetta**: rappresenta l’azienda che distribuisce e pubblicizza un disco, identificata da una partita IVA e da un nome
* **Traccia:** o canzone, presente all’interno del disco, e che presenta un titolo ed una durata
* **Autore:** rappresenta colui che si occupa di cantare o suonare una traccia o, per quanto riguarda il disco, colui che lo ha pubblicato

## Progettazione concettuale

* Riportate qui il **modello ER iniziale**. Cercate di renderlo *leggibile*, altrimenti correggerlo diventerà impossibile. Se è troppo piccolo, dividetelo in parti e/o allegate anche un’immagine ad alta risoluzione alla relazione.



* Commentate gli elementi non visibili nella figura (ad esempio il contenuto degli attributi composti) nonché le scelte/assunzioni che vi hanno portato a creare determinate strutture, se lo ritenete opportuno.
* **Genere** rappresenta un attributo multivalore dal momento che un disco può avere più di un genere.
* **Durata** è un attributo calcolato sulla base della somma delle durate di ogni traccia presente nel disco stesso.
* La **visibilità** rappresenta i due stati in cui può trovarsi un disco:

1. *privato* - ciò significa che una determinata collezione che appartiene ad un collezionista è visibile solo a lui o eventualmente anche agli altri collezionisti con cui è condivisa
2. *pubblica* - ciò significa che la collezione è visibile a tutti i collezionisti

### Formalizzazione dei vincoli non esprimibili nel modello ER

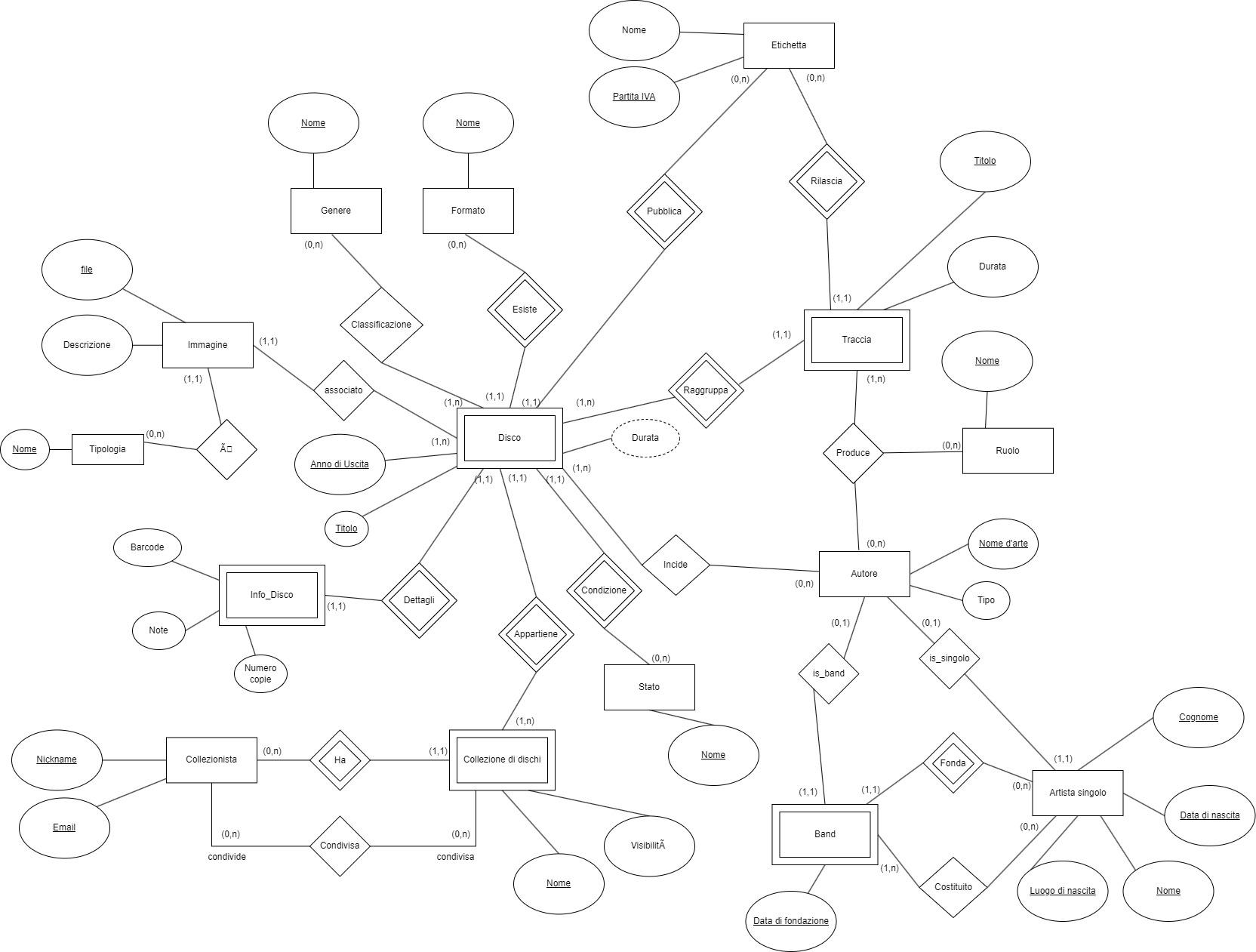
* Elencate gli altri **vincoli** sui dati che avete individuato e che non possono essere espressi nel diagramma ER
* **Collezionista:** *nickname* e *email* sono chiavi primarie dunque non possono essere **NULL.**
* **Collezioni di dischi:** *nome* e gli attributi della relazione *ha* (collezionista) sono chiave primaria dunque non possono essere **NULL.** Il flag *visibilità* non può essere **NULL**, può avere solo *pubblica/privata* come valori.
* **Disco:** *titolo*, *anno di uscita, formato, stato* e la relazione *appartiene*(collezione) e *pubblica* (etichetta) sono la chiave primaria dunque non possono essere **NULL.** Barcode è **UNIQUE** dal momento che deve essere univoco.
* **Tracce:** *titolo* e gli attributi della relazione *rilascia*(etichetta) sono la chiave primaria dunque non possono essere **NULl.** *durata* non può essere **NULL.**
* **Immagine:** *file* è chiave primaria dunque non può essere **NULL.** L’attributo *tipologia* non può essere **NULL.**
* **Etichetta:** *Partita\_IVA* è chiave primaria dunque non può essere **NULL.**
* **Band:** *Data di fondazione* e gli attributi della relazione *fonda*(artista singolo) sono chiave primaria dunque non possono essere **NULL**
* **Artista Singolo:** *nome*, *cognome*, *luogo di nascita*, *data di nascita* sono chiave primaria dunque non possono essere **NULL.**
* **Genere:** *nome* è chiave primaria dunque non può essere **NULL.**
* **Formato:** *nome* è chiave primaria dunque non può essere **NULL.**

.

## Progettazione logica

### Ristrutturazione ed ottimizzazione del modello ER

* Riportate qui il modello **ER ristrutturato** ed eventualmente ottimizzato.



* Discutete le scelte effettuate, ad esempio nell’eliminare una generalizzazione o nello scindere un’entità.

Si è scelto di dividere l’entità Disco in due entità separate (**Disco** ed **Info\_Disco**) per fare in modo che questa non fosse troppo carica di attributi. Inoltre, tale divisione è avvenuta in virtù di quegli attributi meno importanti come, appunto, le *note***,** il *barcode* e il *numero di copie*.

Le generalizzazioni presenti nel modello ER sono state eliminate in favore di due nuove relazioni: **is\_band** e **is\_singolo** tra **Autore**, **Band** e **Artista Singolo** per fare in modo di tenere separate le tre entità senza sovraccaricare le entità figlie o padre

Sono state aggiunte due nuove entità: **Genere** e **Formato**. Precedentemente erano attributi dell’entità disco e sono state slegate da quest’ultima perché, qualora si volesse aggiungere un nuovo genere o formato al database, ciò avrebbe richiesto una modifica strutturale. Pertanto, rendendole entità, tale aggiunta (così come la modifica o la cancellazione) risulta molto meno impattante e facile da eseguire. Infine, genere rappresentava un attributo multivalore che nel modello relazionale si traduce in una nuova entità.

### Traduzione del modello ER nel modello relazionale

* Riportate qui il **modello relazionale** finale, derivato dal modello ER ristrutturato della sezione precedente e che verrà implementato in SQL in quella successiva.

**Collezionista**(ID,nickname,email)

**Collezione\_di\_dischi**(ID ,nome,visibilità, *ID\_collezionista*)

**Disco**(ID, titolo, anno\_di\_uscita, nome\_formato, nome\_stato, *ID\_etichetta*, *ID\_collezione\_di\_dischi*)

**Info\_disco**(ID\_disco, barcode, note, numero\_copie)

**Etichetta**(ID, nome, partita\_IVA)

**Genere**(nome)

**Formato**(nome)

**Ruolo**(nome)

**Stato**(nome)

**Tipologia**(nome)

**Immagine**(ID, file, descrizione, *ID\_disco*, nome\_tipologia)

**Traccia**(ID, titolo, durata, *ID\_etichetta*, *ID\_disco*)

**Autore**(ID, nome\_darte, tipo)

**Artista\_Singolo**(ID, nome, cognome, data\_di\_nascita, luogo\_di\_nascita, *ID\_autore*)

**Band**(ID, ID\_fondatore, data\_di\_fondazione, *ID\_autore*)

**Condivisa**(*ID\_collezione\_di\_dischi*, *ID\_collezionista*)

**Classificazione**(*ID\_disco, nome\_genere*)

**Produce**(*ID\_traccia, ID\_autore, nome\_ruolo*)

**Costituito**(*ID\_band, ID\_membro*)

**Incide**(*ID\_disco, ID\_autore*)

**\***In underline sono evidenziate le primary key mentre in *italic* le foreign key  
Nel modello evidenziate le chiavi primarie e le chiavi esterne.

## Progettazione fisica

### Implementazione del modello relazionale

* Inserite qui lo *script SQL* con cui **creare il database** il cui modello relazionale è stato illustrato nella sezione precedente. Ricordate di includere nel codice tutti i vincoli che possono essere espressi nel DDL.
* Potete opzionalmente fornire anche uno script separato di popolamento (INSERT) del database su cui basare i test delle query descritte nella sezione successiva.

### Implementazione dei vincoli

* Nel caso abbiate individuato dei **vincoli ulteriori** che non sono esprimibili nel DDL, potrete usare questa sezione per discuterne l’implementazione effettiva, ad esempio riportando il codice di procedure o trigger, o dichiarando che dovranno essere implementati all’esterno del DBMS.

### Implementazione funzionalità richieste

* Riportate qui il **codice che implementa tutte le funzionalità richieste**, che si tratti di SQL o di pseudocodice o di entrambi. *Il codice di ciascuna funzionalità dovrà essere preceduto dal suo numero identificativo e dal testo della sua definizione*, come riportato nella specifica.
* Se necessario, riportate anche il codice delle procedure e/o viste di supporto.

#### Funzionalità 1

Definizione come da specifica

CODICE

#### Funzionalità 2

Definizione come da specifica

CODICE

## Interfaccia verso il database

* Opzionalmente, se avete deciso di realizzare anche una **(semplice) interfaccia** (a linea di comando o grafica) in un linguaggio di programmazione a voi noto (Java, PHP, …) che manipoli il vostro database , dichiaratelo in questa sezione, elencando le tecnologie utilizzate e le funzionalità invocabili dall’interfaccia.
* Il relativo codice sorgente dovrà essere *allegato* alla presente relazione.

**Raccomandazioni finali**

* Questo documento è un modello che spero possa esservi utile per scrivere la documentazione finale del vostro progetto di Laboratorio di Basi di Dati.
* Cercate di includere tutto il codice SQL nella documentazione, come indicato in questo modello, per facilitarne la correzione. Potete comunque allegare alla documentazione anche il *dump* del vostro database o qualsiasi altro elemento che ritenete utile ai fini della valutazione.
* Ricordate che la documentazione deve essere consegnata, anche per email, almeno *una settimana prima* della data prevista per l’appello d’esame. Eventuali eccezioni a questa regola potranno essere concordate col docente.