

# Database per la gestione di un archivio di diagrammi ER

## Introduzione

Questo documento contiene le specifiche per la realizzazione di una base di dati il cui scopo è gestire un archivio di diagrammi entità-relazione, insieme al relativo autore e ai permessi di modifica e/o visione per ogni diagramma.

Le varie fasi della costruzione del database verranno illustrate secondo la seguente scaletta:

1. Analisi dei requisiti.
2. Progettazione concettuale.
3. Documentazione
4. Progettazione logica.
5. Progettazione e realizzazione fisica del database tramite DBMS.

## Analisi dei requisiti

La base di dati deve gestire la creazione e la modifica di diagrammi ER da parte di utenti, i cui dati sono a loro volta salvati all'interno del database.

### Dati sugli utenti

Per ogni utente, quindi, va memorizzato un username univoco, una password, il nome anagrafico, un'email ed un recapito telefonico.

Alcuni utenti vengono nominati amministratori del sistema, ottenendo dei poteri esclusivi ed avendo il compito di moderare quest'ultimo.

### Dati sui diagrammi

Per ogni diagramma va memorizzato un identificativo univoco, un nome, le entità e le associazioni che lo compongono, la data di creazione ed infine il suo autore. Inoltre, per esso è necessario gestire i permessi di modifica e visualizzazione da parte di eventuali altri utenti a cui vengono concessi. L'autore, oltre a poter modificare il diagramma, può anche modificare i permessi degli altri utenti rispetto ad esso. Il permesso di modificare implica ovviamente quello di visualizzazione. I

diagrammi posso essere pubblici, ovvero visualizzabili da chiunque possenga un account. In questo caso è necessario registrare solo gli utenti con permesso di modifica.

Per elemento ER si intende un generico elemento contenuto all'interno del diagramma, sia esso un'entità o un'associazione.

Per ogni elemento ER va memorizzato un nome (unico all'interno del diagramma a cui appartiene) e tutti i suoi attributi.

Per un singolo attributo va specificato un nome (unico relativamente all'elemento a cui è legato) e la sua cardinalità.

Per le associazioni va inoltre specificata la loro cardinalità.

Non sono previsti attributi multivalore.

## Note sull'eliminazione di account o diagrammi

Ogni diagramma può essere modificato in ogni sua parte o eliminato. Stesso discorso vale per gli account. Nel caso un account venga eliminato, tutti i diagrammi ER da esso creati vengono a sua volta cancellati. Nel caso di eliminazione di un diagramma o di un account, i permessi riguardanti esso vanno eliminati.

## Glossario

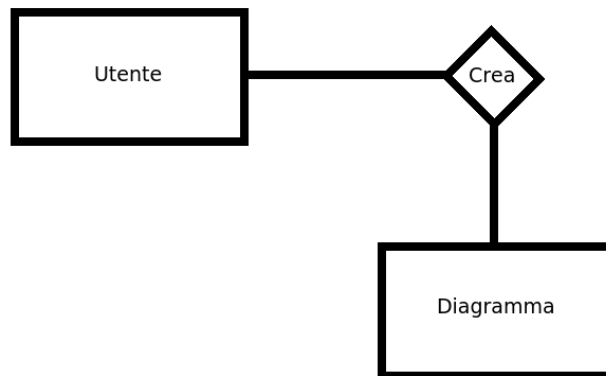
Termine	Descrizione	Sinonimi	Termini legati
Utente	Persona avente accesso al sistema.	User, account	Diagramma, permesso
Amministratore	Utente avente privilegi esclusivi al fine di gestire e moderare il sistema.	Admin	Utente
Diagramma	Diagramma ER salvato all'interno del sistema.	Diagramma ER, Schema ER	Utente, permesso, entità, associazione

Elemento ER	Elemento generico contenuto all'interno di un diagramma ER. Può indicare sia un'entità che un'associazione.		Entità, associazione, diagramma.
Entità	Entità del sistema formale ER contenuta in uno ed un solo diagramma		Diagramma, associazione
Associazione	Associazione del sistema formale ER contenuta in uno ed un solo diagramma, fra 2 o più entità.	Relazione	Diagramma, entità
Attributo	Attributo relativo ad una ed una sola entità o associazione.		Diagramma, entità, associazione
Identificatore	Attributo o insieme di attributi che identificano univocamente un record di un elemento ER nel database rappresentato dal diagramma cui appartiene.	Chiave primaria	Diagramma, entità, associazione
Permesso	Permesso di modifica o visualizzazione da parte di un utente di un diagramma privato	Autorizzazione	Diagramma, utente.

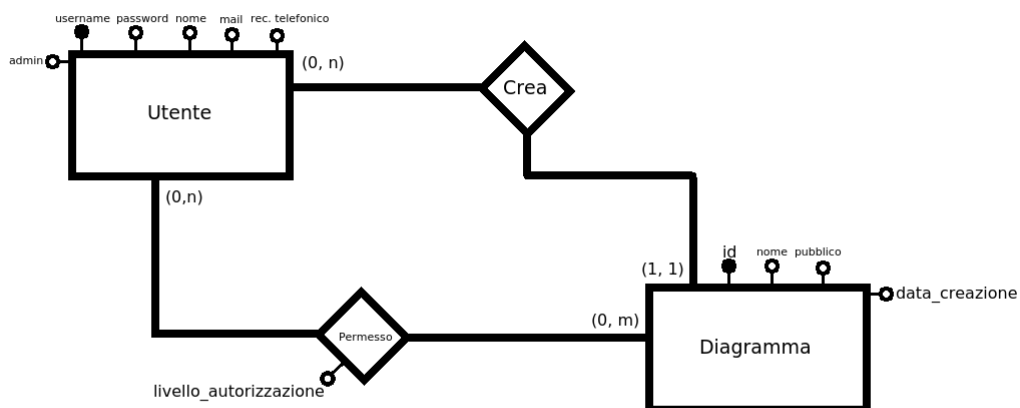
# Progettazione concettuale

Procederemo alla progettazione della base di dati secondo una strategia mista fra bottom-up e top-down, definendo separatamente due schemi. il primo riguardante gli account, i diagrammi e le loro caratteristiche ed i relativi permessi. Il secondo riguarderà le entità e le associazioni che esistono fra loro relative ad un generico diagramma. Sfruttando poi il legame che esiste fra un diagramma e le sue entità ed associazioni uniremo i due schemi formandone un unico compatto.

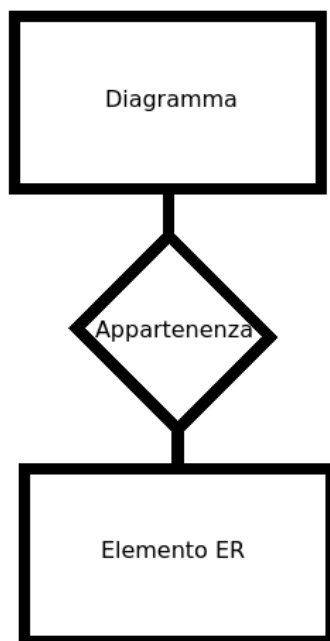
*Schema iniziale riguardante il legame fra utenti e diagrammi:*



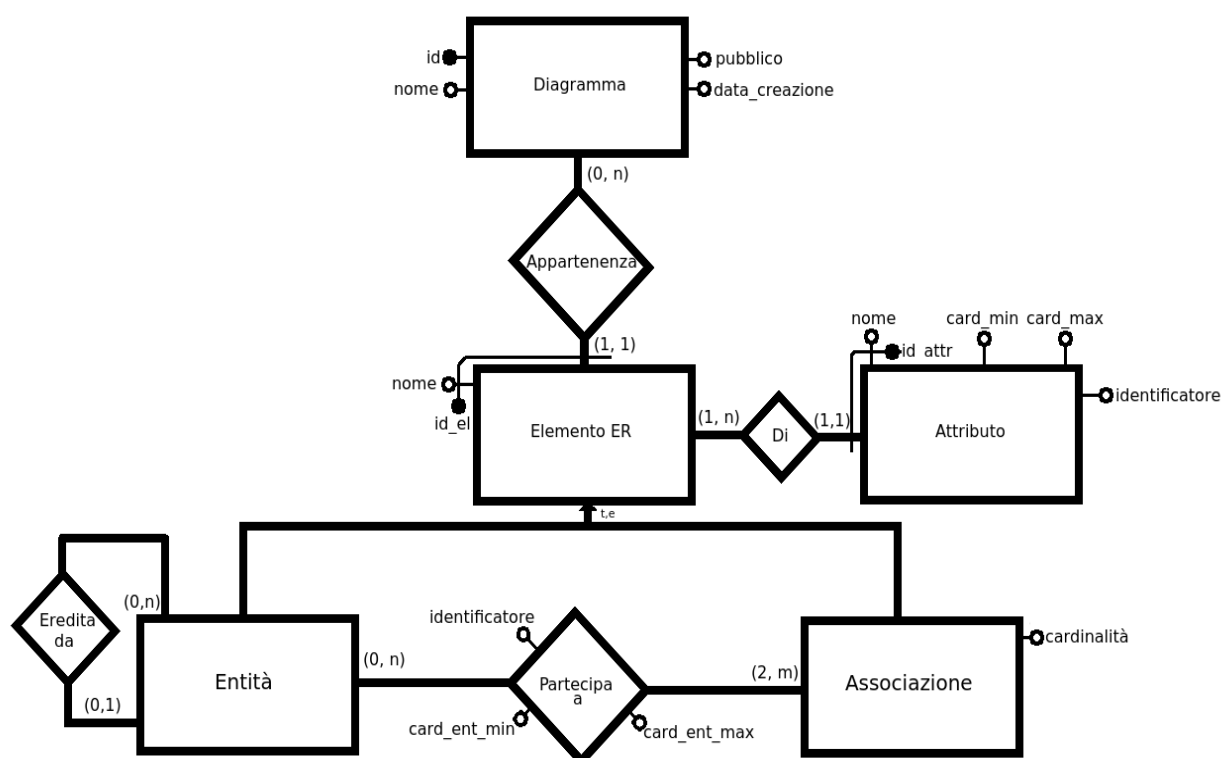
*Schema intermedio riguardante il legame fra utenti e diagrammi:*



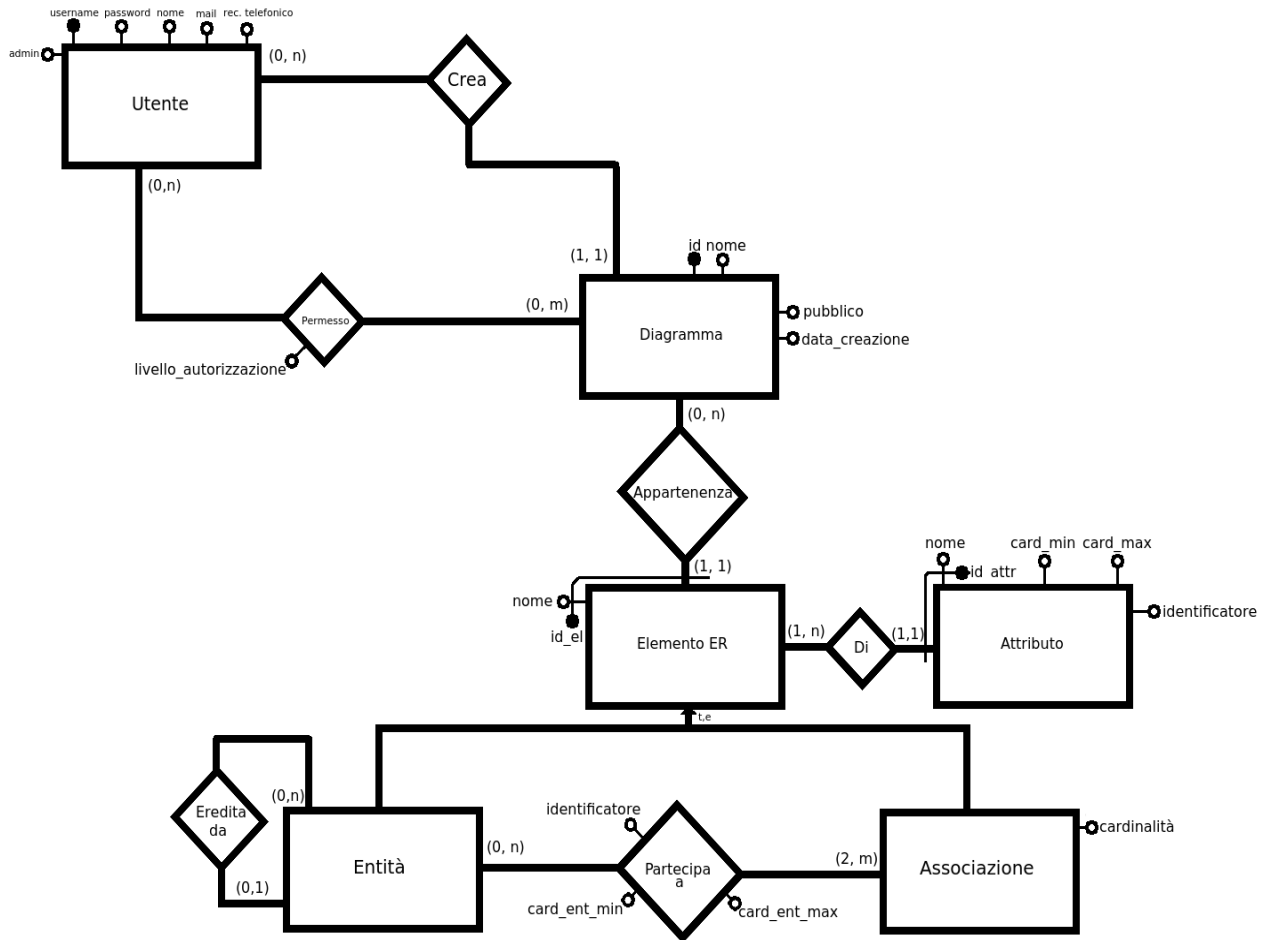
*Schema iniziale riguardante il legame fra diagrammi ed elementi ER:*



*Schema intermedio riguardante il legame fra diagrammi ed elementi ER:*



### Schema finale (non ristrutturato):



## Documentazione

### Dizionario delle entità

Entità	Descrizione	Attributi	Identificatore
Utente	Persona avente accesso al sistema tramite un account registrato	admin, username, nome, mail, rec. telefonico	username
Diagramma	Diagramma ER archiviato nel sistema	id, nome, pubblico, data_creazione	id
Elemento ER	Elemento del modello entità-relazione appartenente ad un diagramma	nome	id_el (tramite la relazione 'appartenenza')

Entità	Entità del modello ER, appartenente ad un diagramma	Vedi 'Elemento ER'	Vedi 'Elemento ER'
Associazione	Associazione del modello ER, appartenente ad un diagramma.	Vedi 'Elemento ER', cardinalità.	Vedi 'Elemento ER'
Attributo	Attributo di un elemento ER appartenente ad un diagramma.	nome, card_min, card_max, identificatore	id_attr (tramite l'attributo 'nome' e la relazione 'Di')

### Dizionario delle relazioni

Relazione	Entità Partecipanti	Descrizione	Attributi
Crea	Utente, Diagramma	Associa ogni diagramma col suo autore	
Permesso	Utente, Diagramma	Registra per ogni diagramma gli utenti autorizzazione a vederlo o modificarlo	livello_autorizzazione
Appartenenza	Diagramma, Elemento ER	Associa ad ogni elemento ER il diagramma a cui appartiene	
Di	Elemento ER, Attributo	Associa ad ogni attributo l'elemento ER a cui partecipa	
Partecipa a	Entità, Associazione	Tiene conto di tutte le entità che partecipano ad una determinata associazione	card_ent_min, card_ent_max, identificatore

Eredita da	Entità, Entità	Associa ogni entità figlia di una gerarchia all'entità padre	
------------	----------------	---	--

## Regole di vincolo

- L'attributo 'card\_ent\_min' della relazione "Partecipa a" dev'essere minore o uguale all'attributo 'card\_ent\_max'. Il discorso è analogo per gli attributi 'card\_min' e 'card\_max' dell'entità "Attributo".
- Le entità possono partecipare solo alle associazioni che appartengono al loro stesso schema. A loro volta, le associazioni possono avere partecipanti solo da entità appartenenti al loro stesso schema.
- Un'entità può avere un rapporto di eredità solo se non eredita già da un'entità, con un'altra entità diversa da se stessa, che appartiene allo stesso diagramma e che non eredita a sua volta dalla prima.
- Un attributo di un'associazione non può far parte del suo identificatore.

## Regole di derivazione

- L'attributo 'cardinalità' dell'entità "Associazione" può essere derivato calcolando il numero di record all'interno della tabella "Partecipa a" relativi all'associazione presa in considerazione.

# Progettazione logica

Il database è pensato per essere utilizzato da un sito web o da un'azienda per memorizzare i diagrammi ER realizzati dai suoi utenti/dipendenti. Per adempire a questo scopo, è necessario implementare le seguenti operazioni (altre possono ovviamente essere considerate, per il fine di questo documento considereremo solo le più frequenti):

- O1: Crea un nuovo diagramma, salvandone l'autore.
- O2: Cambiare il permesso di un utente rispetto ad un diagramma.
- O3: Inserire un nuovo elemento ER in un diagramma.
- O4: Creare un attributo e legarlo ad un elemento ER.
- O5: Aggiungere un rapporto di ereditarietà fra due entità.
- O6: Far partecipare un'entità ad una determinata associazione.
- O7: Calcolare quante entità partecipano ad una determinata associazione.
- O8: Calcolare i diagrammi creati in media da ogni utente.



## Tavola dei volumi

Concetto	Tipo	Volume
Utente	E	100
Diagramma	E	500
Entità	E	4000
Associazione	E	2000
Attributo	E	20000
Crea (Diagramma)	R	500
Permesso	R	400
Appartenenza	R	6000
Di (Attributo)	R	6000
Eredita da (Entità)	R	1000
Partecipa a (Associazione)	R	4500

## Tavola delle frequenze

Operazione	Descrizione	Frequenza	Tipo
O1	Crea un diagramma	15/giorno	I
O2	Cambia un permesso	5/giorno	I
O3	Inserisci un elemento ER	50/ora	I
O4	Crea un attributo	125/ora	I
O5	Aggiungere un'entità ad una gerarchia	15/ora	I
O6	Collegare un'entità ad un'associazione	25/ora	I
O7	Calcolare la cardinalità di un'associazione	15/ora	I

O8	Media dei diagrammi creati per utente	1/settimana	B
----	---------------------------------------	-------------	---

## Analisi delle ridondanze

- L'attributo 'cardinalità' di "Associazione" è ridondante poiché può essere calcolata contando le occorrenze dell'associazione nella relazione "Appartenenza".

Le operazioni O1, O2, O3, O4, O5 sono indipendenti dall'attributo ridondante ed hanno i seguenti costi:

O1:

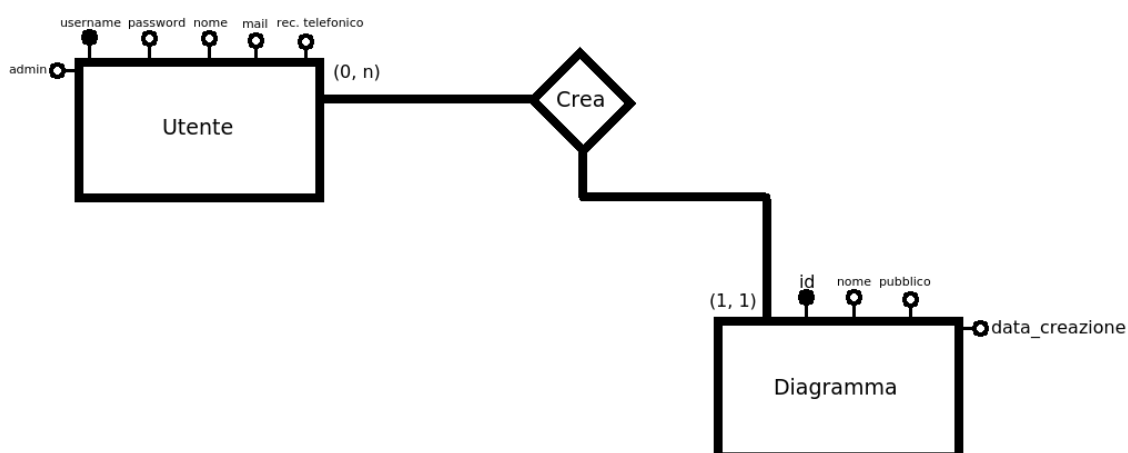


Tabella	Tipo tabella	Accessi	Tipo accesso
Utente	E	1	L
Crea	R	1	S
Diagramma	E	1	S

$$(1L + 2S) * 15/\text{giorno} = 75/\text{giorno}$$

O2:

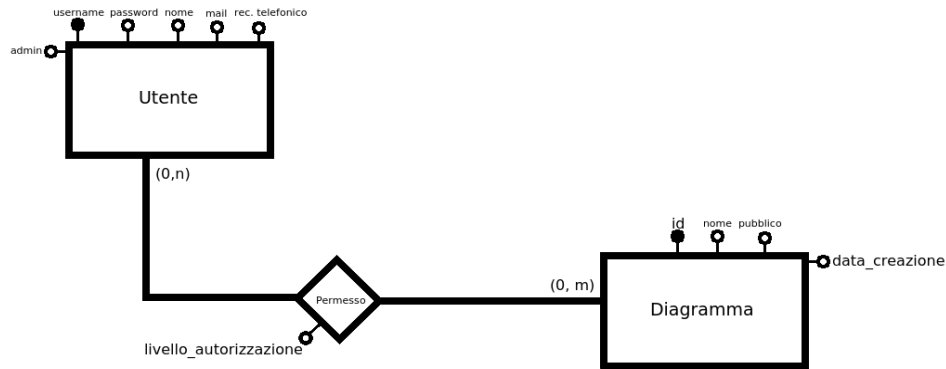


Tabella	Tipo tabella	Accessi	Tipo accesso
Utente	E	1	L
Diagramma	E	1	L
Permesso	R	1	S

$$(2L + 1S) * 5/\text{giorno} = 20/\text{giorno}$$

O3:

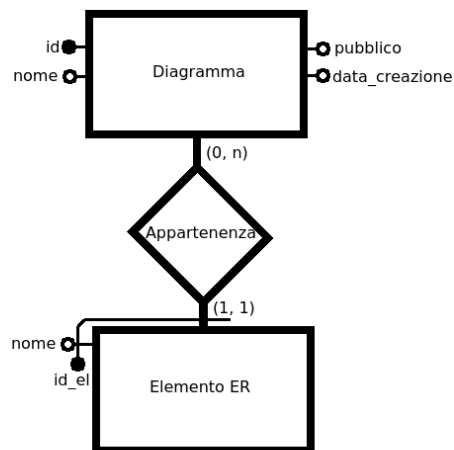


Tabella	Tipo tabella	Accessi	Tipo accesso
Diagramma	E	1	L
Appartenenza	R	1	S
Elemento ER	E	1	S

$$(2S + 1L) * 50/\text{ora} = 250/\text{ora}$$

O4:

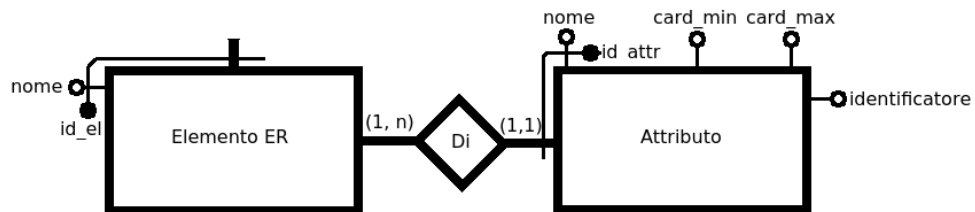


Tabella	Tipo tabella	Accessi	Tipo accesso
Elemento ER	E	1	L
Di (Attributo)	R	1	S
Attributo	E	1	S

$$(1L + 2S) * 125/ora = 750/ora$$

O5:

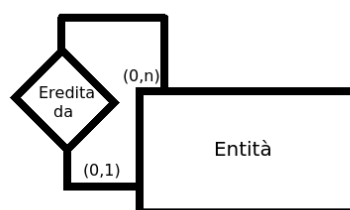


Tabella	Tipo tabella	Accessi	Tipo accesso
Entità	E	2	L
Eredita da	R	1	S

$$(2L + 1S) * 15/ora = 60/ora$$

L'operazione **O8** è indipendente dall'attributo ridondante ed inoltre è un'operazione batch. Essa ha il seguente costo:

**O8:**

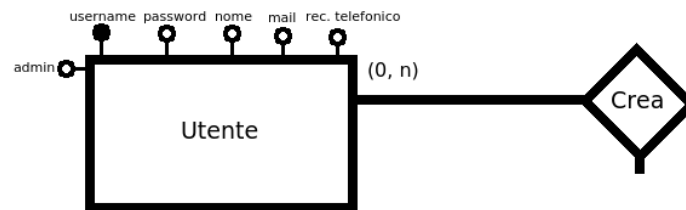


Tabella	Tipo tabella	Accessi	Tipo accesso
Utente	E	1 * 100 utenti	L
Crea (Diagramma)	R	5 * 100 utenti	L

$$(6L) * 100 * 1/\text{settimana} = 600/\text{settimana}$$

Le operazioni **O6** ed **O7** sono invece legate all'attributo ridondante e di conseguenza ne valutiamo i costi mantenendo o eliminando la ridondanza:

**Con ridondanza:**

**O6:**

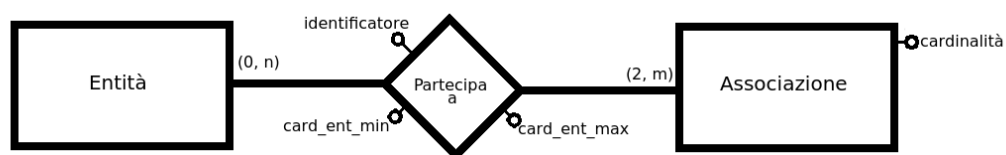


Tabella	Tipo tabella	Accessi	Tipo accessi
Associazione	E	1	L
Entità	E	1	L
Partecipa a	R	1	S
Associazione	E	1	S

$$(2L + 2S) * 25/\text{ora} = 150/\text{ora}$$

O7:

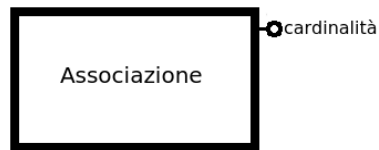


Tabella	Tipo tabella	Accessi	Tipo accessi
Associazione	E	1	L

$$(1L) * 15/ora = 15/ora$$

Costo totale: 140/ora

Senza ridondanza:

O6:

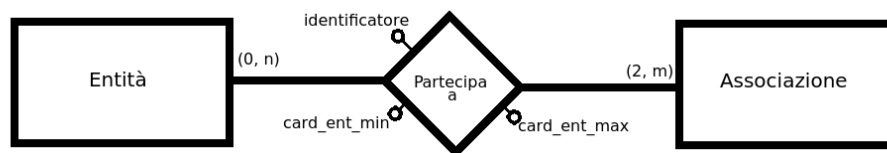


Tabella	Tipo tabella	Accessi	Tipo accessi
Associazione	E	1	L
Entità	E	1	L
Partecipa a	R	1	S

$$(2L + 1S) * 25/ora = 100/ora$$

O7:

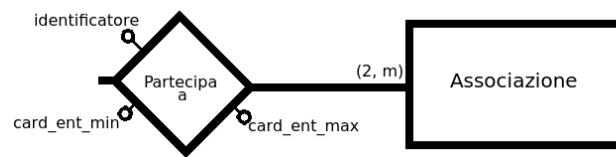


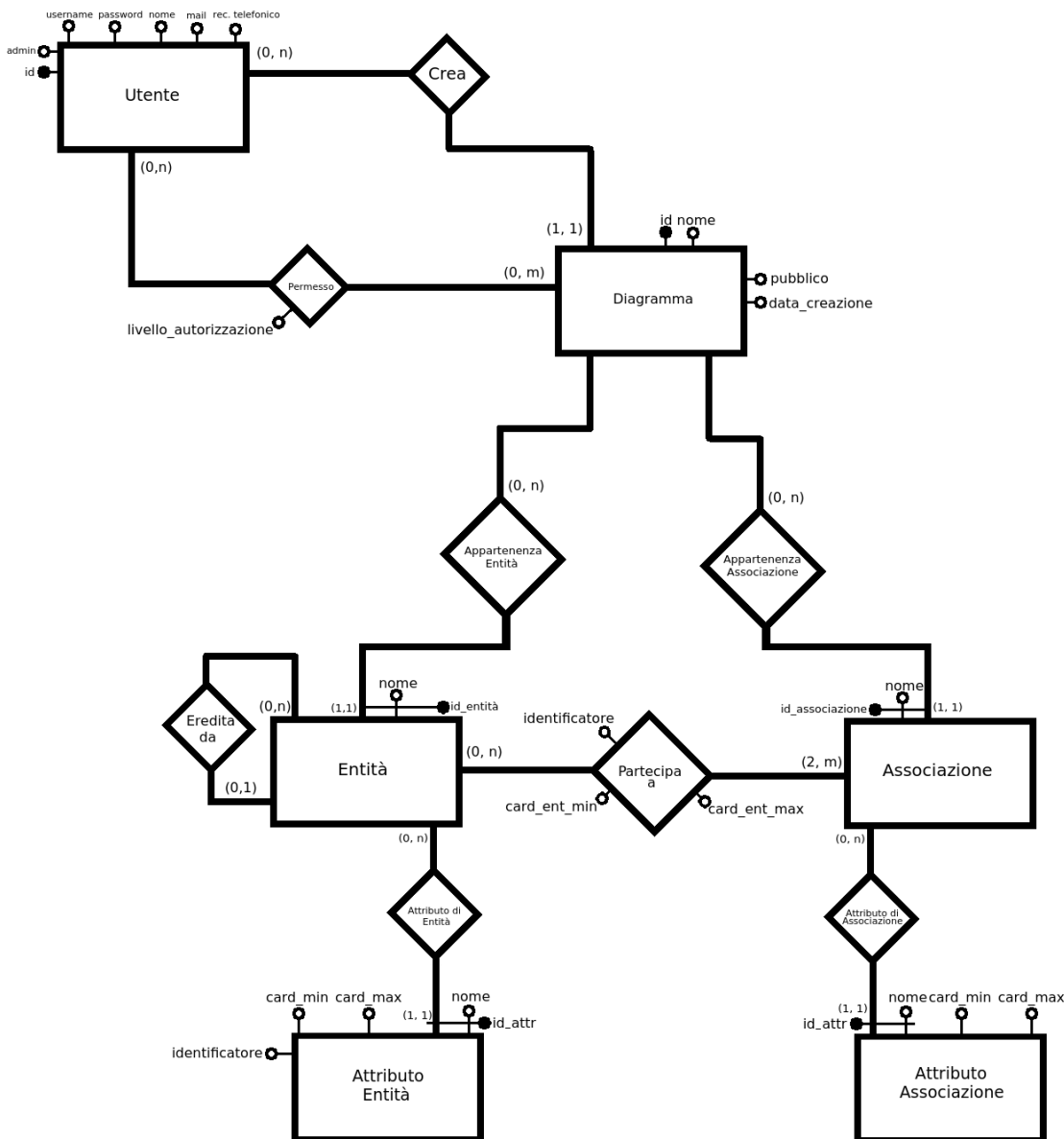
Tabella	Tipo tabella	Accessi	Tipo accessi
Associazione	E	1	L
Partecipa a	R	3 (arrotondati da 2.5)	L

$$(4L) * 15 / \text{ora} = 60/\text{ora}$$

Costo totale: 135/ora

Supponendo che l'attributo cardinalità necessiti di 4 byte per record per essere registrato, è opportuno **eliminare la ridondanza**, poiché non solo essa richiede più operazioni per essere aggiornata rispetto ad essere calcolata quando è necessario consumerebbe anche  $4 * 2000 = 8000$  byte in più per essere memorizzata.

Procediamo adesso a ristrutturare lo schema ER:



Traducendolo nel modello relazionale si ottiene il seguente schema:

Utente(id, username, password, nome, mail, rec\_telefonico, isAdmin)  
 Diagramma(id, nome, pubblico, id\_autore, data\_creazione)  
 Permisso(id\_utente, id\_diagramma, livello)  
 Entita(id\_entità, nome, id\_diagramma)  
 Associazione(idAssociazione, nome, id\_diagramma)  
 AttributoEntita(nome, id\_entità, isIdentificatore, card\_min, card\_max)  
 AttributoAssociazione(nome, idAssociazione, card\_min, card\_max)  
 PartecipazioneAssociazione(id\_entità, idAssociazione, isIdentificatore, card\_ent\_min, card\_ent\_max)  
 Ereditarieta(id\_ent\_figlia, id\_ent\_padre)



## Vincoli d'integrità dello schema ristrutturato

- L'attributo 'card\_ent\_min' della relazione "PartecipazioneAssociazione" dev'essere minore o uguale all'attributo 'card\_ent\_max' della stessa relazione. Il vincolo è analogo per gli attributi 'card\_min' e 'card\_max' delle relazioni "AttributoEntita" e "AttributoAssociazione".
- Le entità possono partecipare solo alle associazioni che appartengono al loro stesso schema. A loro volta, le associazioni possono avere entità partecipanti solo da entità appartenenti al loro stesso schema.
- Un diagramma non può avere due entità o due associazioni con lo stesso nome.
- Un'entità può avere un rapporto di eredità solo se non eredita già da un'entità, con un'altra entità diversa da se stessa, che appartiene allo stesso diagramma e che non eredita a sua volta dalla prima.
- All'eliminazione dell'account di un utente vanno eliminati tutti i diagrammi da lui creati e tutti i permessi legati ad esso.
- All'eliminazione di un diagramma vanno eliminati tutti gli elementi ER che conteneva e tutti i permessi relativi ad esso.
- All'eliminazione di un elemento ER vanno eliminati tutti i suoi attributi e tutti i record ad esso legati nelle tabelle PartecipazioneAssociazione.
- All'eliminazione di un'entità vanno eliminate tutti i record ad essa legati nella tabella Ereditarieta.

## Progettazione Fisica

Per l'implementazione della base di dati è stato scelto un DBMS relazionale, MariaDB.

Sono state inserite, oltre alle 8 operazioni descritte in questo documento, alcune procedure che consentono modifiche o cancellazioni verosimilmente necessarie al completo funzionamento del DB, pur mantenendo lo scopo di essere un semplice esempio e non una base di dati pronta ad essere utilizzata in uno specifico ambiente di produzione.

In allegato:

Immagini utilizzate all'interno del documento.

Versione in formato .ODT del documento.

La cartella "sql" contiene i seguenti file per l'inizializzazione ed un primo debug del database:

"tables.sql": Crea le tabelle del database.

"triggers.sql": Crea i trigger del database.

"procedures.sql": Implementa alcune operazioni comuni nel database e delle funzioni ausiliarie.

"init.sql": Inizializza il database.

"testdata.sql": Inserisce alcuni dati di test nel database. Alcuni violano volontariamente i vincoli imposti e sono utilizzati per testare la funzionalità di quest'ultimi.

"testout.sql": Stampa a schermo il contenuto del database e l'output di alcune funzioni. Utile per verificare l'integrità della base di dati.