

Errori ricorrenti nella progettazione concettuale

Questo documento ha, come scopo, presentare alcuni gravi errori che ricorrono spesso nella progettazione concettuale di basi di dati effettuata dagli studenti del corso, nelle prove d'esame. Svolgeremo la parte di progettazione concettuale del compito d'esame dell'8 settembre 2012, soffermandoci in particolare su:

- le cardinalità delle relazioni, a volte errate nei compiti di esame;
- altri gravi errori ricorrenti.

Gli errori sono evidenziati in rosso, e poi spiegati. E' molto importante comprendere il motivo per cui gli elementi marcati in rosso sono sbagliati; in caso di dubbi persistenti, è necessario chiedere chiarimenti prima di sostenere l'esame.

Esame dell'8 settembre 2012

Progettare la base di dati a servizio di un sito di condivisione di fotografie. Si vogliono rappresentare le seguenti informazioni:

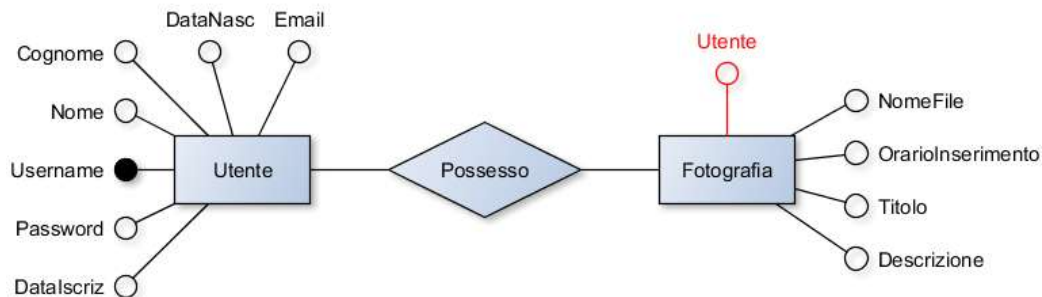
- Gli **utenti** del sito, ciascuno dotato di credenziali di accesso (username e password), nome, cognome, indirizzo email, data di nascita, data di iscrizione al sito;
- Le **fotografie** inserite nel sito. Di ogni foto vogliamo rappresentare l'utente proprietario della foto, il nome del file che contiene la foto, l'orario di inserimento, il titolo, la descrizione;
- I **commenti** di ogni foto, di cui vogliamo rappresentare il testo del commento, l'autore (che è un utente del sito), e l'orario di inserimento;
- I **tag** di ogni foto. Un tag sta ad indicare la presenza nella foto di un utente del sito; in particolare, si vogliono rappresentare l'utente taggato e la posizione nella foto in cui l'utente taggato appare (coordinate x e y del "punto centrale" dell'utente taggato).
- I **blocchi**: ciascun utente può bloccare altri utenti. (Gli utenti bloccati non potranno inserire commenti alle foto o taggare l'utente che li ha bloccati.)

Svolgimento della progettazione concettuale

Iniziamo a rappresentare gli utenti e le fotografie: si tratta di due entità. Esse sono legate da una relazione che chiameremo **Possesso**, che esprime il fatto che le foto appartengono agli utenti:



Ora inseriamo gli attributi richiesti dalle specifiche:

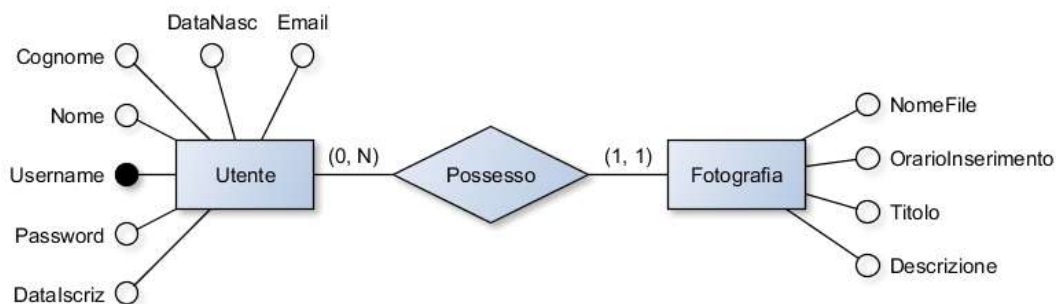


Errore ricorrente 1

Fotografia non deve avere un attributo chiamato **Utente**, in quanto il proprietario della foto è già stato rappresentato nel nostro schema concettuale. La rappresentazione avviene attraverso la relazione **Possesso**, che appunto lega le fotografie agli utenti che ne sono proprietari.

Eliminiamo dunque l'attributo errato, e determiniamo la cardinalità della relazione **Possesso**:

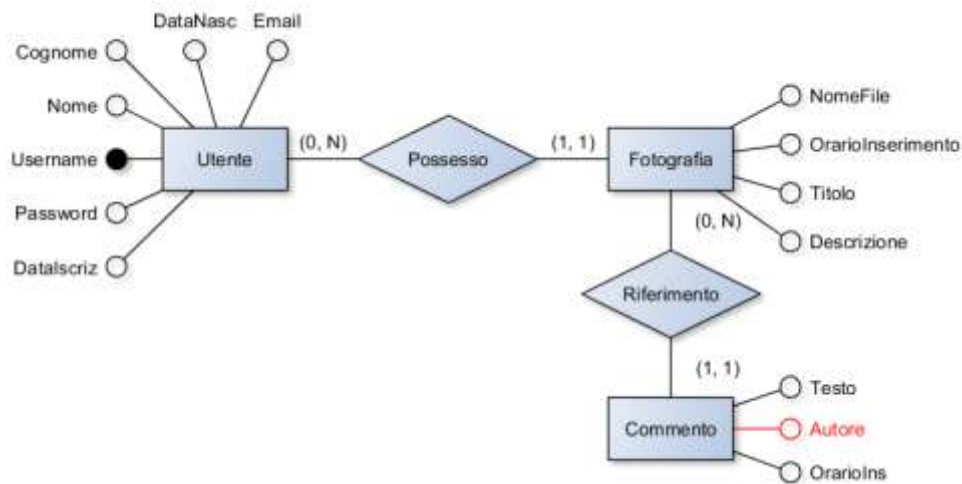
1. (Da **Utente** verso **Fotografia**): ogni utente, quante fotografie possiede? Minimo 0, massimo senza limite -> (0, N)
2. (Da **Fotografia** verso **Utente**): ogni foto, quanti utenti ha come proprietari? Minimo 1, massimo 1 -> (1, 1):



Osserviamo che la cardinalità (1, 1) della partecipazione dell'entità **Fotografia** alla relazione **Possesso** ci dice che ogni foto ha uno e un solo proprietario. Dunque è un po' come se la foto avesse un attributo "proprietario": ecco spiegato il motivo che potrebbe portare a commettere l'errore ricorrente 1.

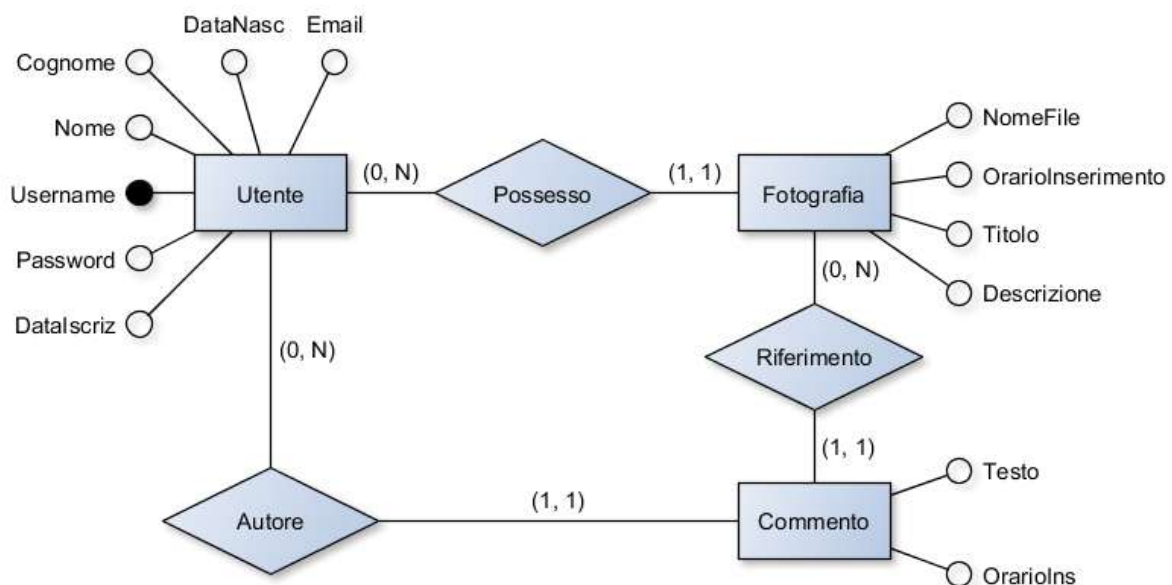
Si osservi inoltre che, durante la successiva progettazione logica, la relazione **Possesso** sarà rappresentata proprio introducendo un attributo aggiuntivo **Proprietario** alla tabella **Fotografie**: pertanto l'attributo che avevamo rimosso dallo schema concettuale perché errato ritornerà ad esistere nello schema logico (però sarà una chiave esterna verso la tabella **Utenti**).

Proseguiamo la progettazione concettuale rappresentando il concetto di **Commento** secondo quanto richiesto dalle specifiche (rileggerle prima di proseguire):



Errore ricorrente 2

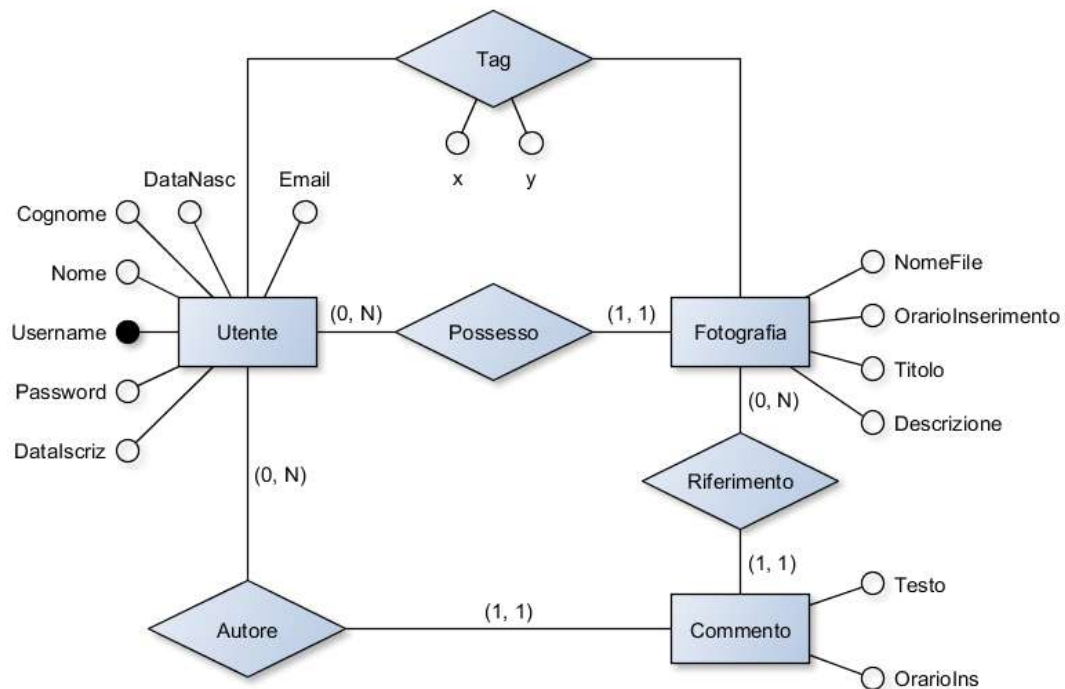
Da un punto di vista concettuale, l'autore di un commento è un utente del sito. Poiché il concetto di utente del sito è presente nello schema entità-relazione, ed ha dignità di entità (infatti esiste l'entità **Utente**), è errato rappresentare l'autore di un commento attraverso un semplice attributo. Bisognerà invece rappresentarlo attraverso una relazione tra **Commento** e **Utente** (avendo cura di rimuovere l'attributo per non ricadere nell'errore ricorrente 1).



Osserviamo, ancora una volta, che **Commento** partecipa alla relazione **Autore** con cardinalità (1, 1), proprio perché ogni commento ha uno e un solo utente come autore (e quindi è un po' come se l'autore fosse un attributo di **Commento**).

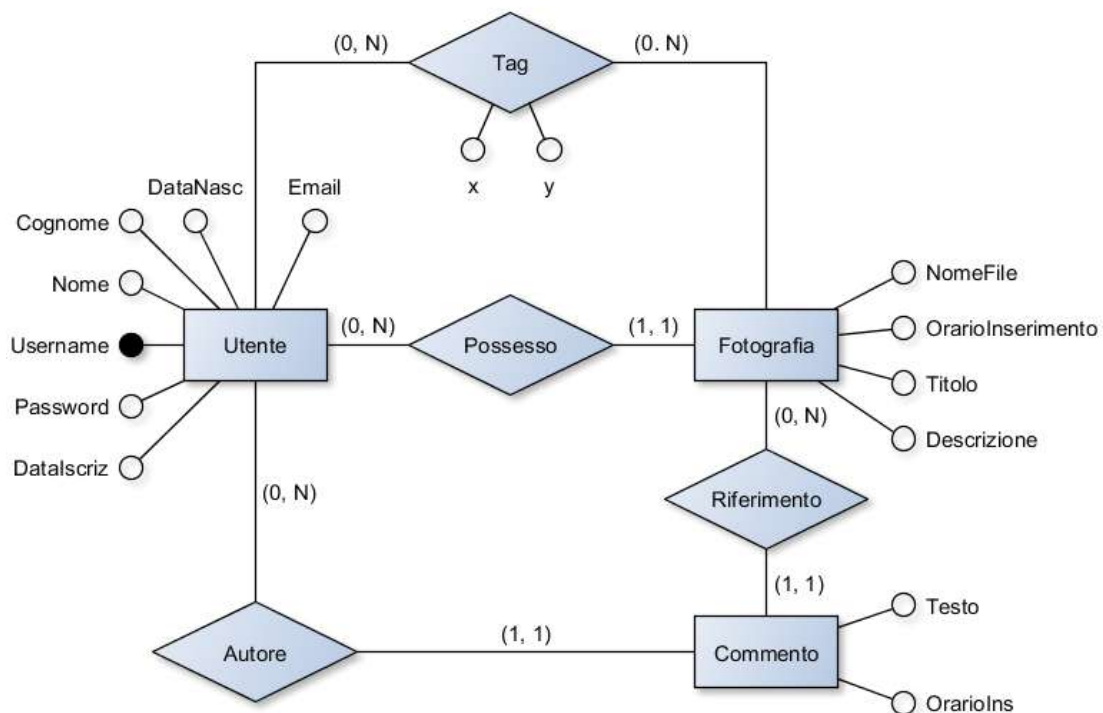
Passiamo ora a rappresentare il concetto di tag. Sarebbe del tutto corretto rappresentarlo attraverso un'entità **Tag** (del tutto analoga a **Commento**), collegata, con due relazioni, alle entità **Utente** (utente taggato) e **Fotografia** (foto in cui appare il tag).

In alternativa, poiché ogni utente può essere taggato al massimo una volta in ciascuna foto, è possibile rappresentare il concetto di Tag mediante una relazione:

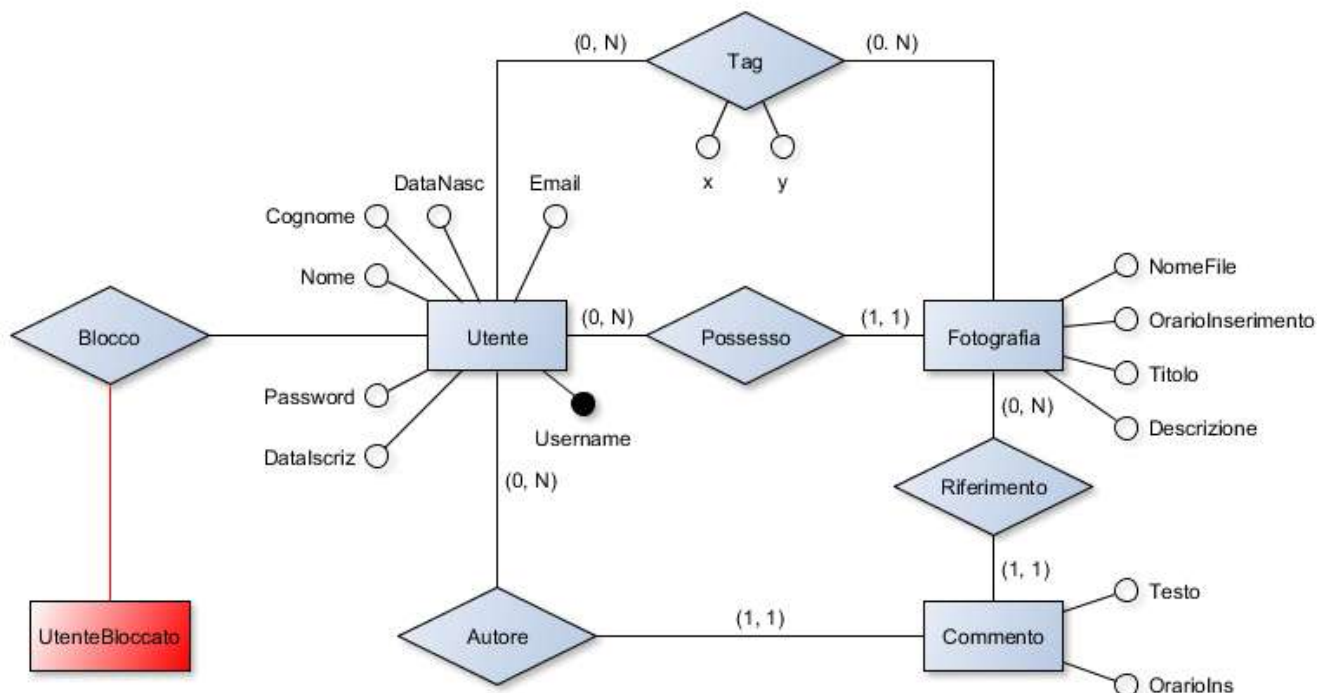


Determiniamo la cardinalità della relazione **Tag**:

1. (Da **Utente** verso **Fotografia**): ogni utente, in quante foto può essere taggato? Minimo 0, massimo senza limite -> (0, N)
2. (Da **Fotografia** verso **Utente**): in ogni foto, quanti utenti possono essere taggati? Minimo 0, massimo senza limite -> (0, N):



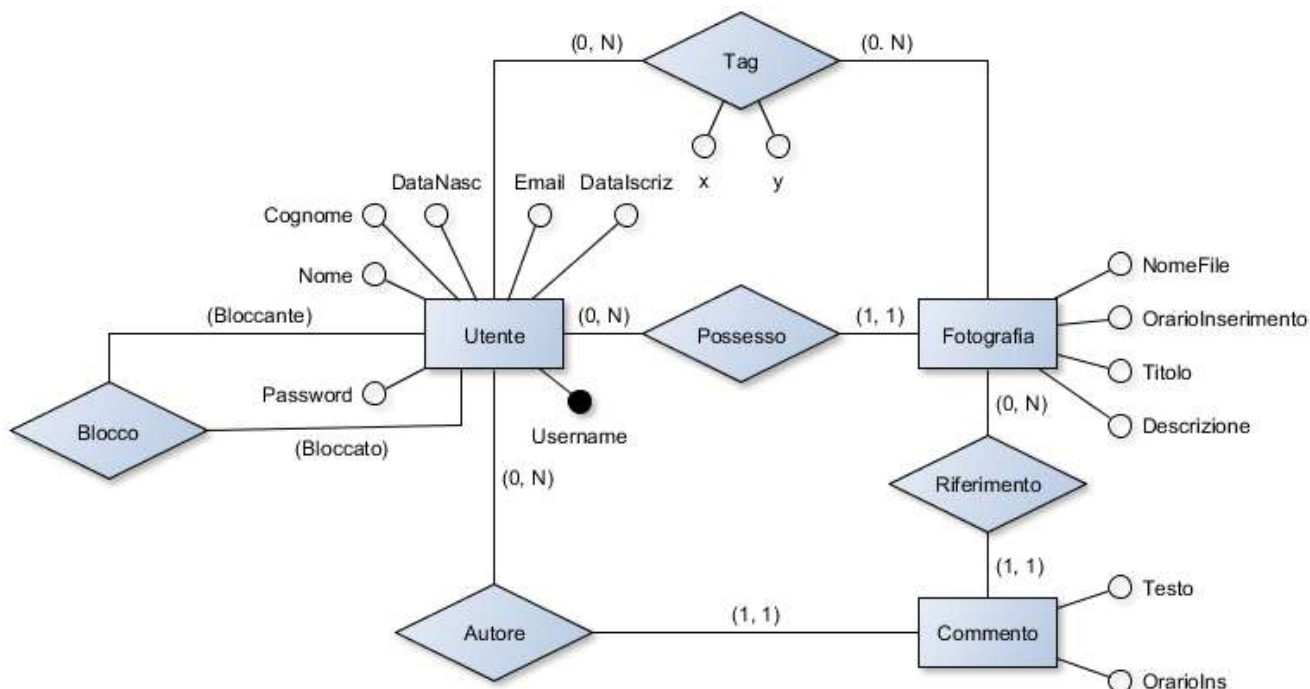
Infine, occorre rappresentare il concetto di blocco. Rileggendo le specifiche completiamo lo schema E-R:



Errore ricorrente 3

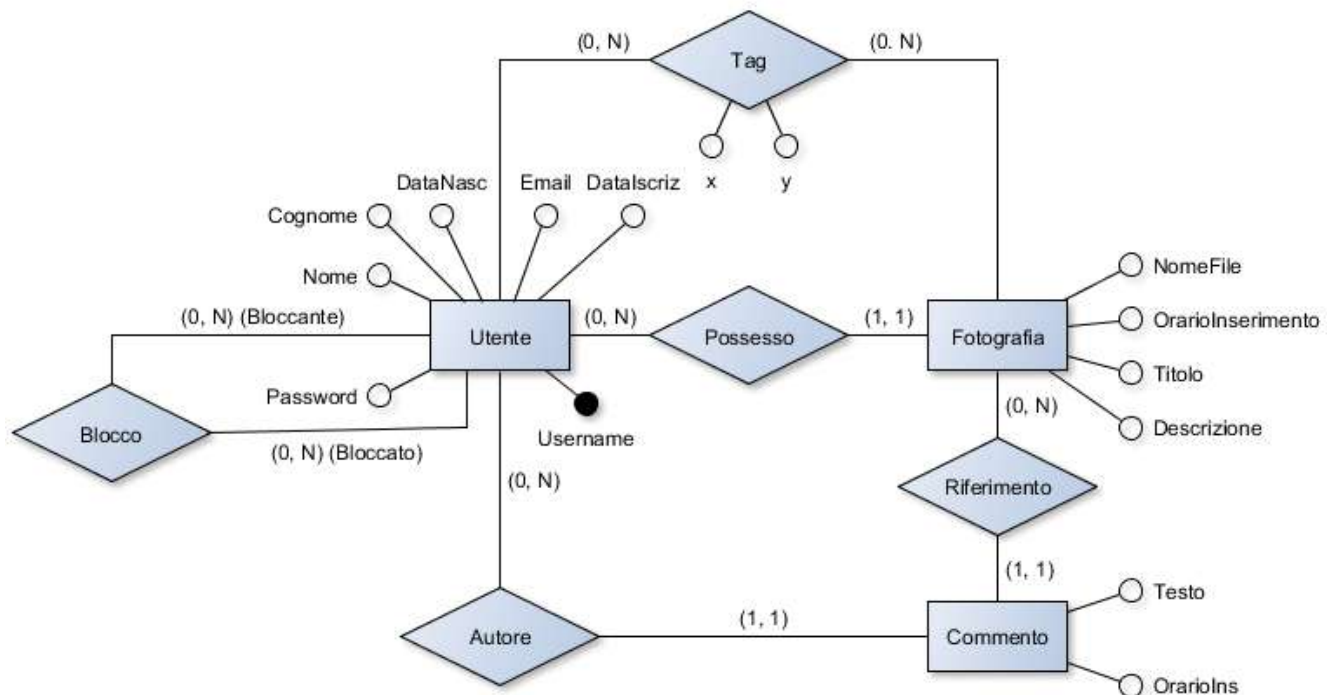
L'utente bloccato altri non è che un utente del sito. In altre parole, il concetto di **Utente** è già presente nello schema entità-relazione: è un errore molto grave introdurre un'altra "copia" del concetto.

In questo caso, **Blocco** è semplicemente una relazione tra un utente (l'utente che dispone il blocco) e un altro utente (l'utente bloccato):



Si noti che abbiamo specificato il ruolo con cui **Utente** partecipa alla relazione **Blocco**: una volta con il ruolo di **Bloccante**, una volta con il ruolo di **Bloccato**. Determiniamo la cardinalità della relazione:

1. (Da **Utente bloccante** verso **Utente bloccato**): ogni utente, quanti altri utenti può bloccare? Minimo 0, massimo senza limiti -> (0, N)
2. (Da **Utente bloccato** verso **Utente bloccante**): ogni utente, da quanti altri utenti può essere bloccato? Minimo 0, massimo senza limiti -> (0, N):



Notiamo che, ancora una volta, sarebbe stato del tutto corretto rappresentare **Blocco** con un'entità. In tal caso avremmo dovuto inserire **due** relazioni tra le entità **Blocco** e **Utente**, che avremmo potuto chiamare **UtenteBloccante** e **UtenteBloccato**. Tali relazioni sarebbero state di tipo uno-a-molti (verificarlo per esercizio).

Morale della favola

Quando un concetto è rappresentato in uno schema mediante un'entità A, tutte le volte che si deve far riferimento ad esso in un'altra entità B, lo si deve fare mediante una relazione, ed essa soltanto. E' quindi errato:

- Introdurre, oltre alla relazione, anche un attributo dell'entità B (errore ricorrente 1);
- Introdurre, al posto della relazione, un attributo dell'entità B (errore ricorrente 2);
- introdurre una "copia" dell'entità A per metterla in relazione con l'entità B (errore ricorrente 3).