

Requisiti per il Database

December 6, 2025

1 Analisi

Si vuole descrivere la realtà del sito di un negozio di pesci, tramite il quale si possono acquistare i suddetti pesci. L'utente che visita il sito potrà diventare cliente, registrandosi (fornendo informazioni base, come nome, cognome, email e scegliendo uno username) e le volte seguenti accedendo, tramite username e password. Il cliente potrà acquistare i pesci, e ogni acquisto avrà la corrispondente quantità di pesci e la data (data+ora) in cui è stato effettuato (si potrebbe mettere anche l'attributo prezzo totale, ma è un attributo ridondante che nel nostro caso non verrà molto coinvolto nelle query, quindi non conviene). Un utente potrebbe anche essere un amministratore che può inserire nuovi pesci arrivati in negozio (un amministratore non ha attributi in più rispetto a un utente, quindi non ha senso fare un'entità a parte). Di ogni pesce si vuole conoscere: nome, famiglia, habitat, alimentazione (carnivoro, erbivoro onnivoro), se è d'acqua dolce o salata (informazione che serve più che altro per facilitare le query, quando vorremo ottenere tutti i pesci d'acqua dolce o salata), la dimensione, il prezzo e una descrizione generale. Alcuni di questi attributi possiamo decidere di toglierli e scriverli dentro la descrizione, tenendo a parte solo quelli fondamentali che vogliamo mostrare subito all'utente, che poi cliccherà sull'immagine del pesce per leggere la descrizione che non è visibile fin da subito (non del tutto almeno), nella mia idea.

Spiegazione della relazione Inserimento

Se usiamo il nome del pesce come chiave primaria, allora è necessario anche un attributo disponibilità, che serve per stabilire quanti esemplari di quel pesce ci sono nel negozio. Così però diventa un po' più complesso lo schema ER, nello specifico la relazione tra le entità Utente e Pesce. Infatti l'amministratore può

creare una nuova riga per inserire un pesce di cui non c'era nessun esemplare nel negozio, oppure può inserire un nuovo pesce arrivato di cui c'erano già degli esemplari in negozio, che corrisponde ad aumentare l'attributo disponibilità.

Così nella relazione Inserimento avrei che un amministratore può inserire da 0 a N pesci, e un pesce può essere inserito da 1 a N amministratori, perché un amministratore può inserire un pesce di cui il negozio non aveva esemplari (nuova riga nella tabella Pesce) e questo accade solo una volta. Ma può anche inserire un nuovo pesce di cui ci sono già degli esemplari in negozio, e quindi aumenta la disponibilità e basta, senza aggiungere righe alla tabella Pesce, e questo può essere fatto più volte sulla stessa riga.

Quindi alla fine la relazione Inserimento diventa una tabella in dove avrò come attributi la mail dell'amministratore (chiave primaria in Utente) e il nome del pesce (chiave primaria in Pesce), ma questa non è una chiave, in quanto lo stesso amministratore potrebbe inserire più volte lo "stesso pesce" nelle modalità sopra spiegate. Quindi dovrei aggiungere un ID come chiave primaria di Inserimento. L'alternativa è usare un ID per ogni esemplare di pesce come chiave primaria, evitando di usare l'attributo disponibilità. Ma concettualmente l'attributo ridondante disponibilità (è ridondante perché se ho un ID per ogni esemplare di pesce, potrei contare quelli con lo stesso nome per ottenerlo) è utile perché la disponibilità è un'informazione che viene richiesta in tutte le query che vengono fatte per ottenere i dati dei pesci e mostrarli agli utenti.

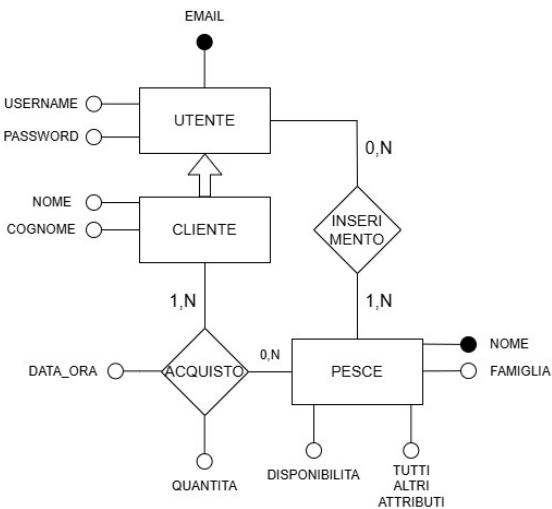


Figure 1: Bozza ER