ARCHITECTURAL PATTERN

Table Data Gateway

Table Data Gateway

E' un oggetto che funge da gateway per una tabella di database.

Un'istanza di un «Data Gateway» gestisce tutte le righe della tabella.

La combinazione di SQL nella logica dell'applicazione può causare diversi problemi

L'oggetto contiene tutto l'SQL utile per accedere a una singola tabella o vista (selects, inserts, updates, and deletes).

I metodi di questo tipo di oggetto vengono chiamati per tutte le interazioni con il database (da cui Gateway).

Table Data Gateway

Esso dispone di un'interfaccia semplice, solitamente costituita da diversi metodi di ricerca per ottenere dati dal database e i metodi per aggiornare, inserire ed eliminare record.

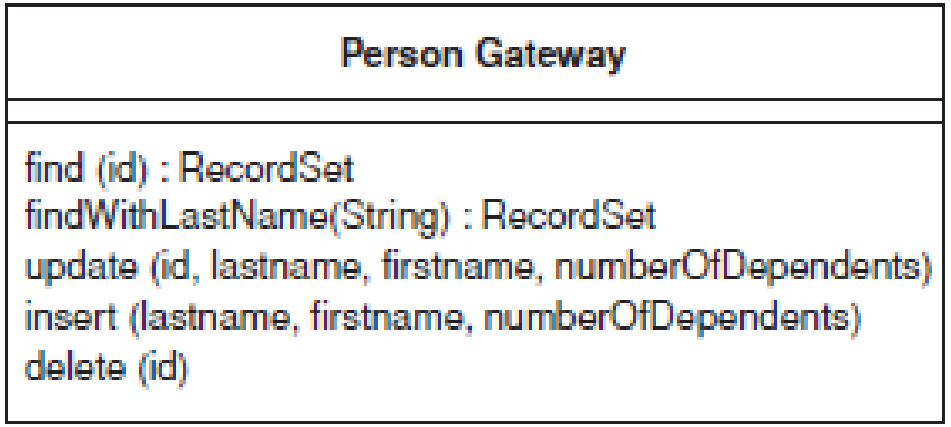


Table Data Gateway

Ogni metodo mappa i parametri di input in una chiamata SQL ed esegue l'SQL su una connessione al database.

E' solitamente senza stato poiché il suo ruolo non è «amministrare» i dati, ma semplicemente riceverli e inviarli.

Ciò che bisogna gestire della fine dei conti è il modo in cui restituire i dati provenienti dalle query.

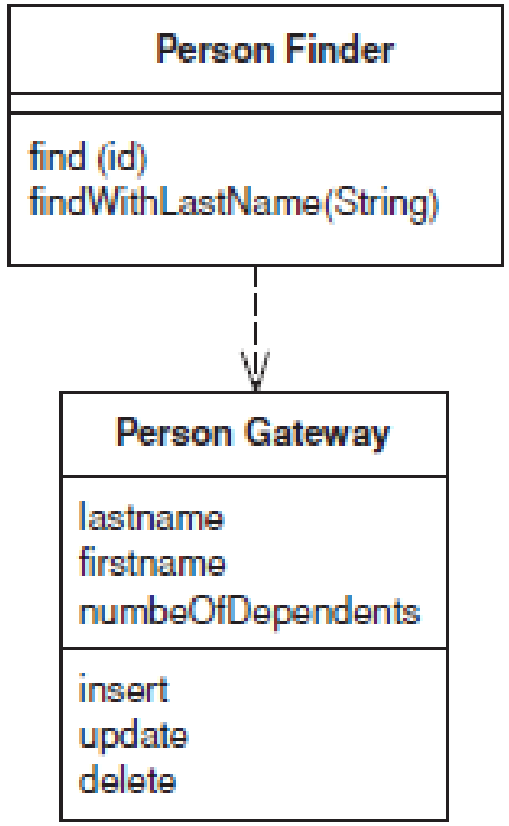
Il metodo migliore è restituire DTO (Data Transfer Object) (o collection di DTO).

Row Data Gateway

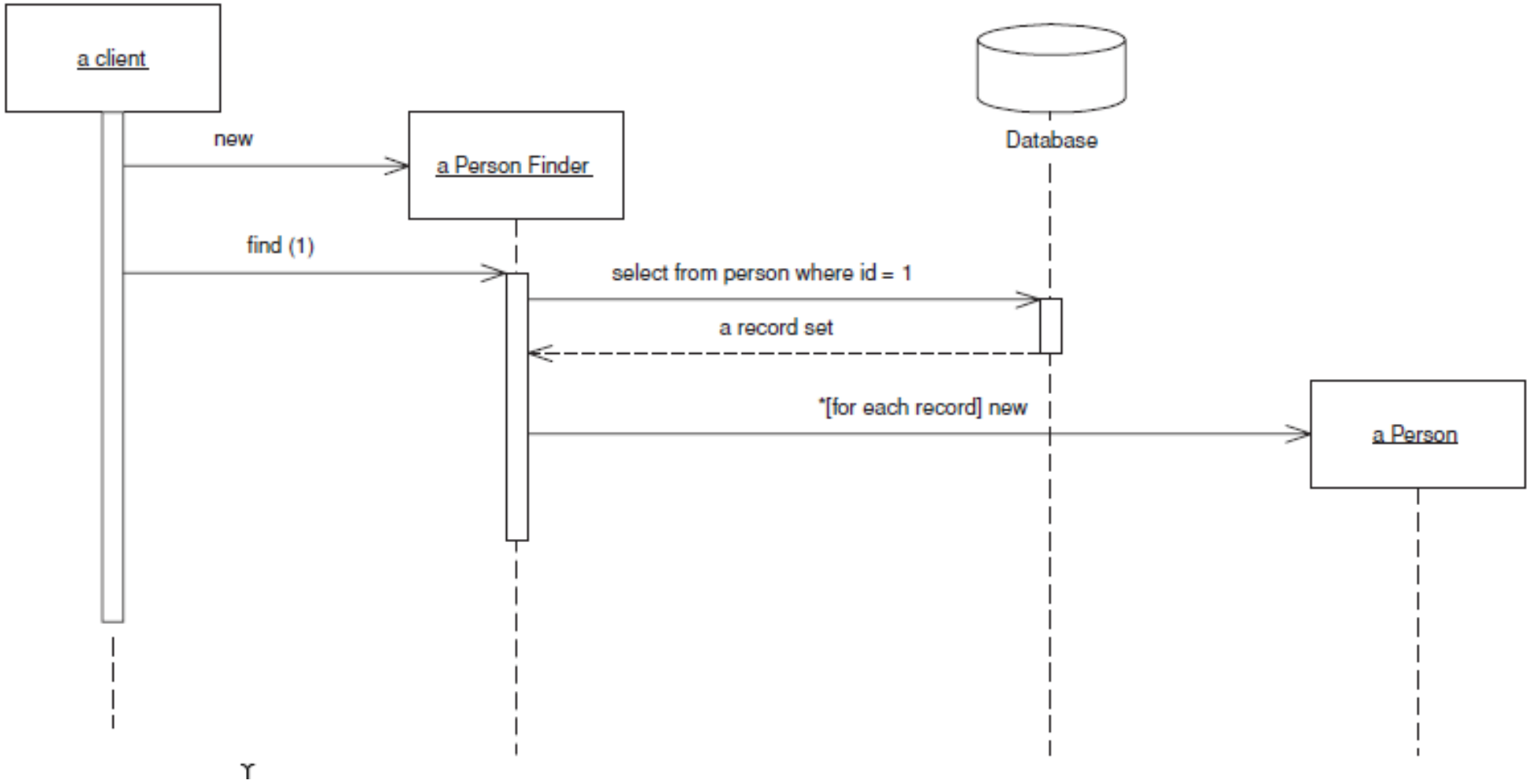
Row Data Gateway

E' un oggetto che funge da gateway per un singolo record di un'origine dati.

Viene creata un'instanza per ogni riga di dati estratti da una tabella o una vista.



Esempio di interazione con un ricerca specializzata su di un Row Data Gateway.

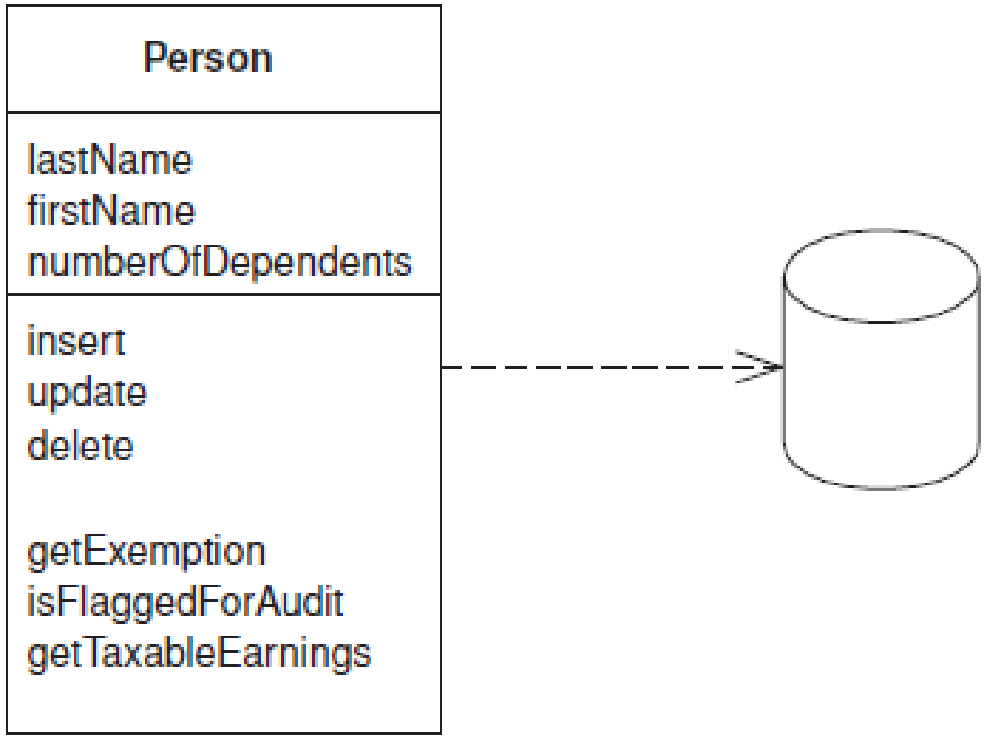


Active Record

Active Record

E' un oggetto che funge da wraps su di una riga di una tabella o di una vista

Incapsula l'accesso al database e aggiunge la logica di dominio ai dati che gestisce.



Active Record

Questo tipo di oggetto gestisce sia dati sia comportamento.

Molti di questi dati sono persistenti e devono essere archiviati in un database.

Active Record utilizza l'approccio più ovvio, inserendo la logica di accesso ai dati nell'oggetto di dominio.

In questo modo si definisce come leggere e scrivere i propri dati da e verso il database.

L'essenza di un Active Record è un Domain Model in cui le classi corrispondono molto strettamente alla struttura del record di una tabella del database sottostante.

Ogni Active Record è responsabile del salvataggio e del caricamento nel database e anche di qualsiasi logica di dominio che agisce sui dati.

L'oggetto potrebbe incapsulare tutta la la logica di dominio, oppure una parte potrebbe essere contenuta in un Transaction Scripts e un'altra (orientata ai dati) nell'Active Record.

La struttura dei dati dell'Active Record deve corrispondere esattamente a quella del database: un campo nella classe per ogni colonna nella tabella.

Active Record

La classe Active Record generalmente possiede metodi che eseguono le seguenti operazioni:

- **Costruire un'istanza dell'Active Record a partire da una riga del set di risultati SQL**
- **Costruire una nuova istanza per l'inserimento dati in tabella**
- **Metodi di ricerca statici per il wrapping sulle query SQL e il ritorno dei dati**

Gli Oggetti Active Record (e quindi le istanze della classe) hanno come responsabilità:

- **Aggiornare il database e inserisci in esso i dati nell'Active Record**
- **Ottenere e gestire i dati di un record.**
- **Implementare alcune parti della logica di business.**

Active Record

Active Record è molto simile al Row Data Gateway.

La differenza principale è che un Row Data Gateway contiene solo l'accesso al database mentre un Active Record contiene sia l'origine dati che la logica del dominio.

Come in altri pattern, la differenza tra i due è una molto sottile, ma comunque utile.

Active Record è una buona scelta per un Domain Logic non troppo complesso.

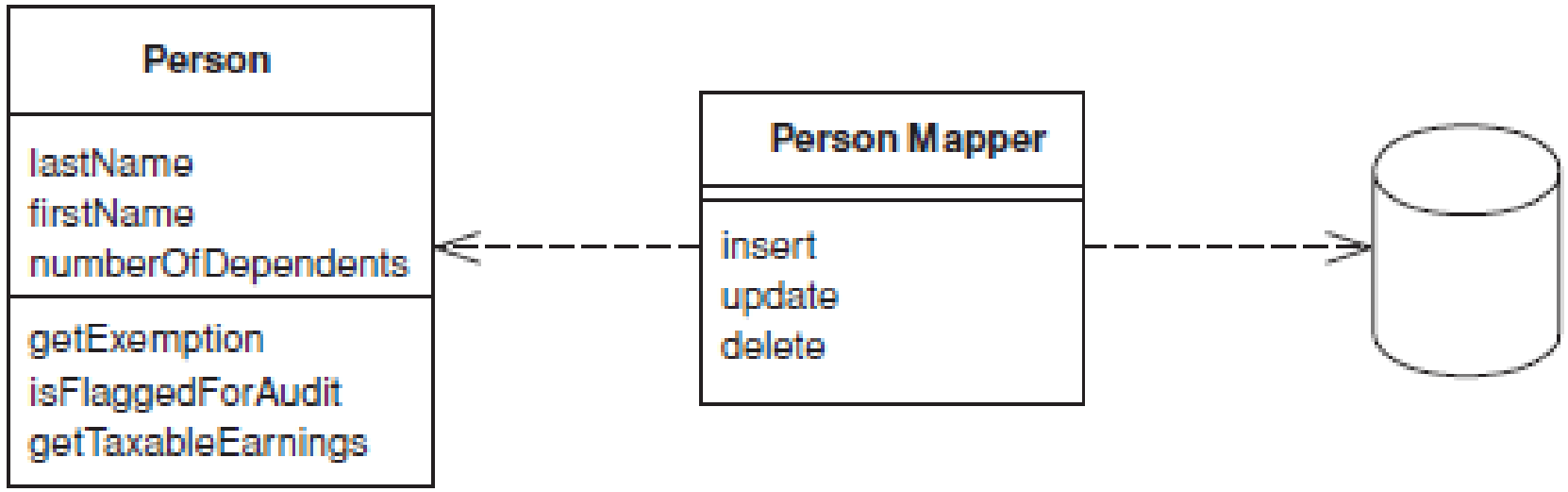
Active Record

Il loro problema principale è che funzionano bene solo se gli oggetti Active Record corrispondono direttamente alle tabelle del database: uno schema isomorfo.

Un altro argomento contro Active Record è il fatto che accoppia l'object design con il database design.

Data Mapper

Un layer di Mapper muove i dati tra gli oggetti e un database mantenendoli indipendenti l'uno dall'altro e indipendenti dal Mapper stesso.



Data Mapper

Oggetti e database relazionali hanno meccanismi diversi per strutturare i dati.

Molte parti di un oggetto, come le collections e l'ereditarietà, non sono presenti nei database relazionali.

Quando si crea un modello a oggetti con molta logica di business, è utile utilizzare questi meccanismi per organizzare meglio i dati e il comportamento che ne consegue.

In questo modo si ottengono schemi varianti; ovvero, lo schema dell'oggetto e lo schema relazionale non corrispondono.

È ancora necessario trasferire i dati tra i due schemi e questo trasferimento di dati diventa una complessità a sé stante.

Data Mapper

Se gli oggetti in memoria conoscono la struttura del database relazionale, i cambiamenti in uno tendono a propagarsi all'altro.

Il Data Mapper è un livello di software che separa gli oggetti in memoria dal database.

La sua responsabilità è trasferire i dati tra i due e anche isolarli l'uno dall'altro.

Con Data Mapper gli oggetti in memoria non devono nemmeno sapere che è presente un database; non hanno bisogno di SQL e certamente nessuna conoscenza dello schema del database.

La separazione tra dominio e origine dati è la funzione principale di un Data Mapper.

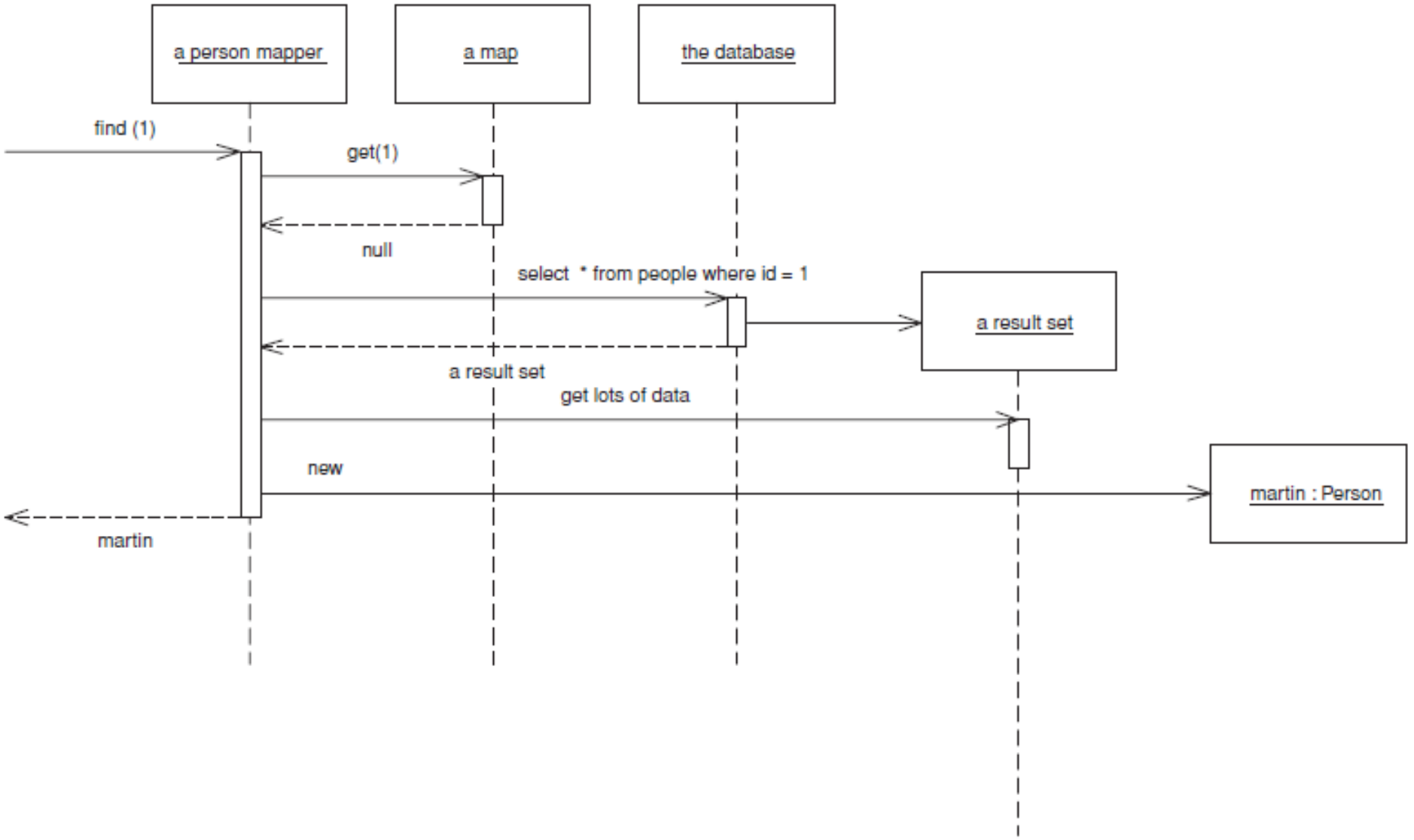
Data Mapper

Quando si tratta di inserimenti e aggiornamenti, il livello di mappatura del database deve comprendere quali oggetti sono cambiati, quali nuovi sono stati creati e quali sono stati distrutti.

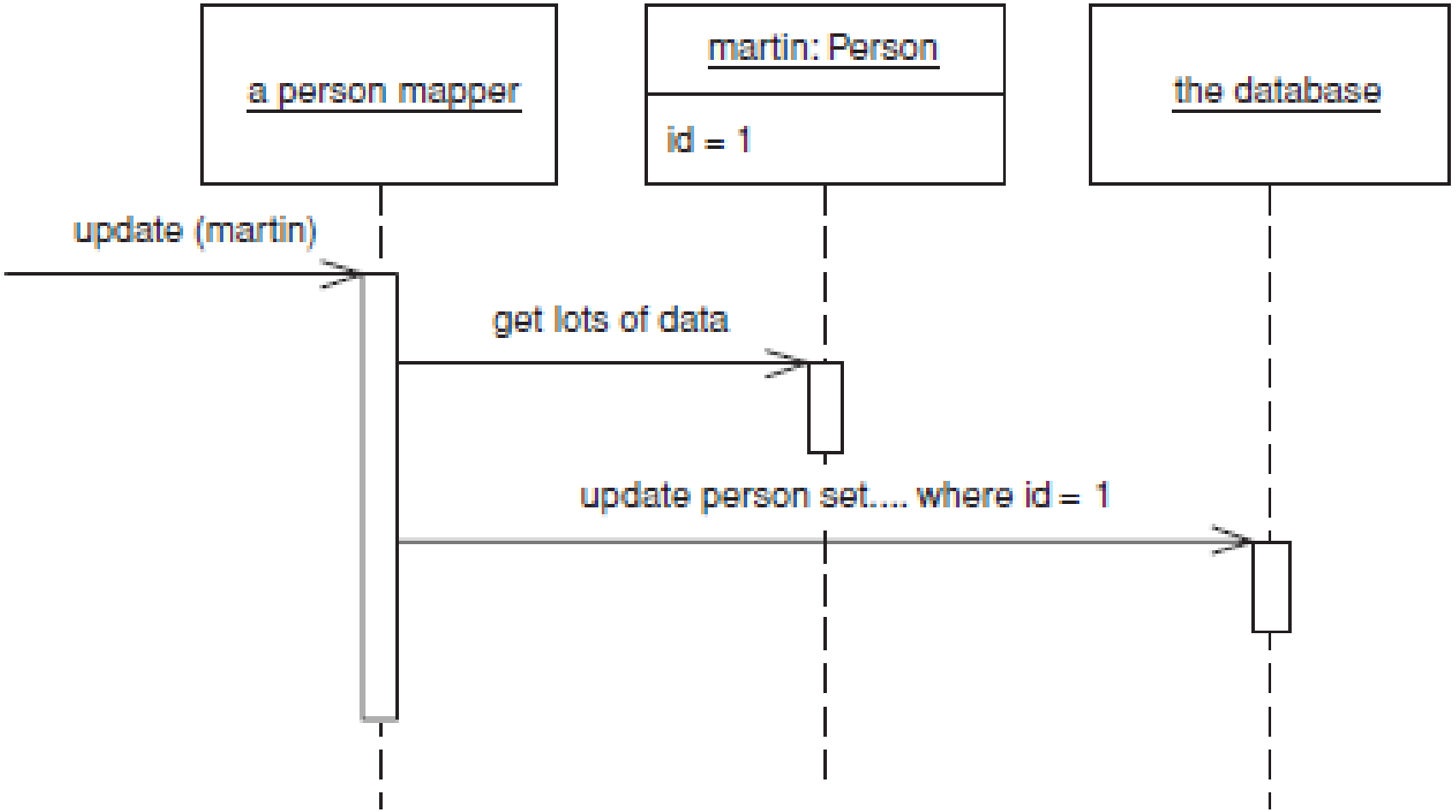
Inoltre deve anche adattare l'intero carico di lavoro in un framework transazionale.

Il problema più grande di questo pattern è come effettuare il caricamento dei dati nella struttura che mappa la tabella o la vista del database.

Esempio di recupero dati da un databse.



Esempio di aggiornamento.



Uso di una interfaccia Finder nel domain package