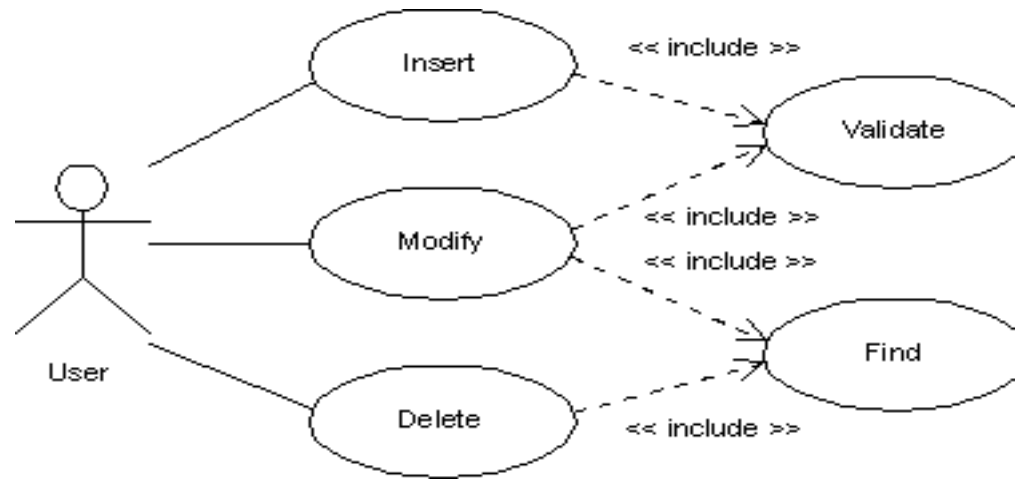

Progettazione e Sviluppo dello strato di Adattamento Object-Relational di un'Applicazione

- **Obiettivi**
 - Realizzare un'applicazione che accede ad una base di dati
 - La base di dati conterrà informazioni idrometriche
 - Realizzare uno strumento per il Mapping O/R
- **Vincoli**
 - La nuova applicazione deve essere compatibile con i vecchi dati
 - La base di dati vecchia esiste in due versioni
 - Database relazionale
 - Sistema di Legacy (file ASCII)
- **Parleremo di**
 - Layering
 - Basi di dati relazionali
 - Modelli ad oggetti
 - Mapping O/R



Use Case 1: Inserisci dati

Primary Actor: User

Scope: System

Level: User-Goal

Main Success Scenario:

1. User crea i nuovi dati da inserire nel database
2. User chiede di archiviare i nuovi dati
3. Il sistema valida i dati
4. Il sistema archivia i dati nel db

Extensions:

- 1-2a. User annulla l'inserimento:
 - 1-2a1. Successo
- 3a. I dati non sono validi:
 - 3a1. Il sistema segnala l'errore
 - 3a2. Riprendi dal punto 1
- 4a. Errore nel salvataggio:
 - 4a1. Il sistema segnala l'errore
 - 4a2. Fallimento

Use Case 2: Modifica dati

Primary Actor: User

Scope: System

Level: User-Goal

Main Success Scenario:

1. User cerca i dati da modificare
2. User modifica i dati
3. User chiede di salvare le modifiche
4. Il sistema valida i dati
5. Il sistema archivia i dati nel db

Extensions:

- 1a. Nessun dato trovato:
 - 1a1. Successo
- 2-3a. User annulla la modifica:
 - 2-3a1. Successo
- 4a. I dati non sono validi:
 - 4a1. Il sistema segnala l'errore
 - 4a2. Riprendi dal punto 2
- 5a. Errore nel salvataggio:
 - 5a1. Il sistema segnala l'errore
 - 5a2. Fallimento

Use Case 3: Cancella dati

Primary Actor: User

Scope: System

Level: User-Goal

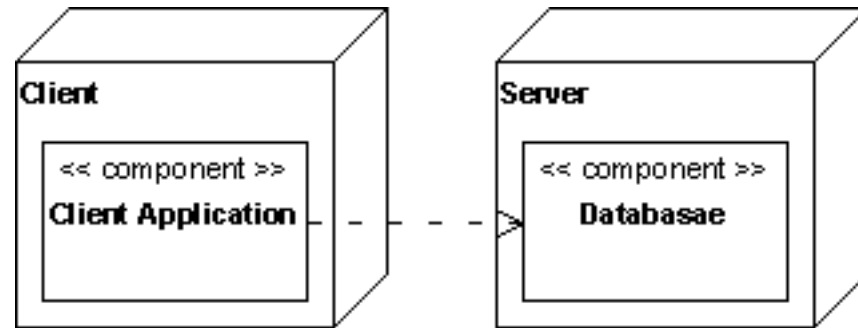
Main Success Scenario:

1. User cerca i dati da cancellare
2. User chiede di cancellare
3. Il sistema chiede conferma
4. Il sistema cancella i dati db

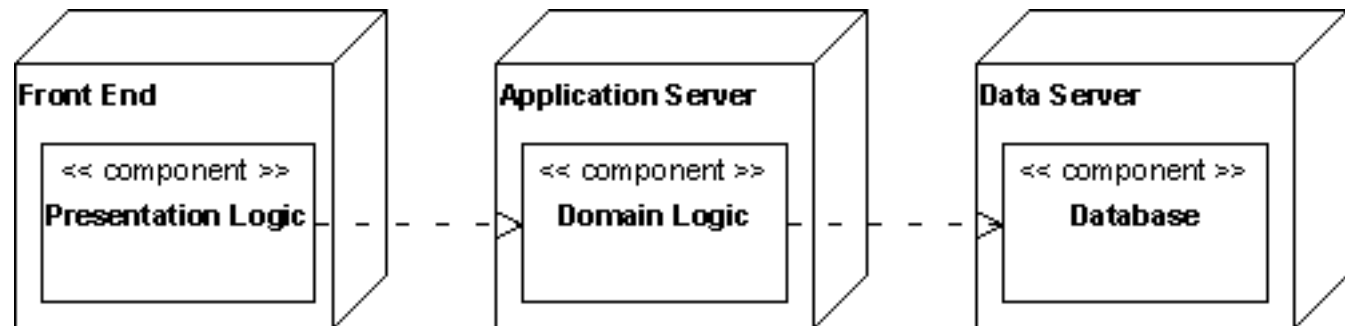
Extensions:

- 1a. Nessun dato trovato:
 - 1a1. Successo
- 2-3a. User annulla la cancellazione:
 - 2-3a1. Successo
- 4a. Errore nella cancellazione sul database
 - 4a1. Il sistema segnala l'errore
 - 4a2. Fallimento

- 2 - Tier (o client/server)

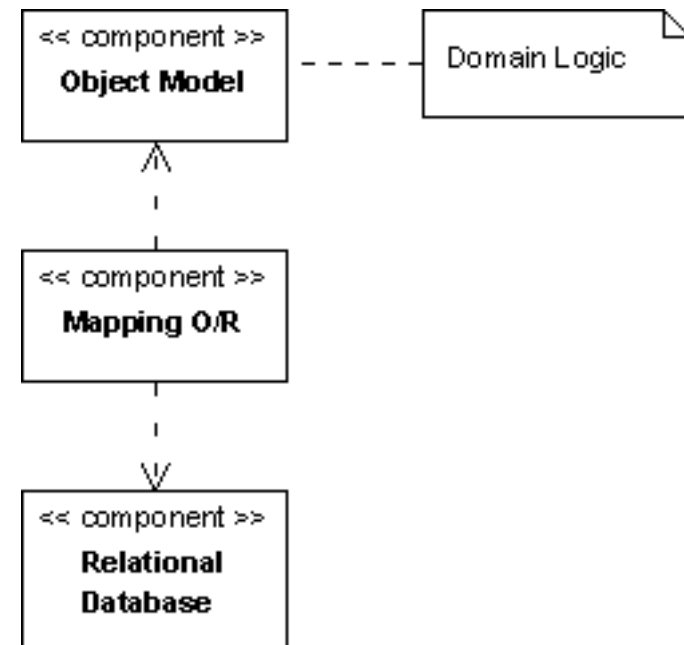


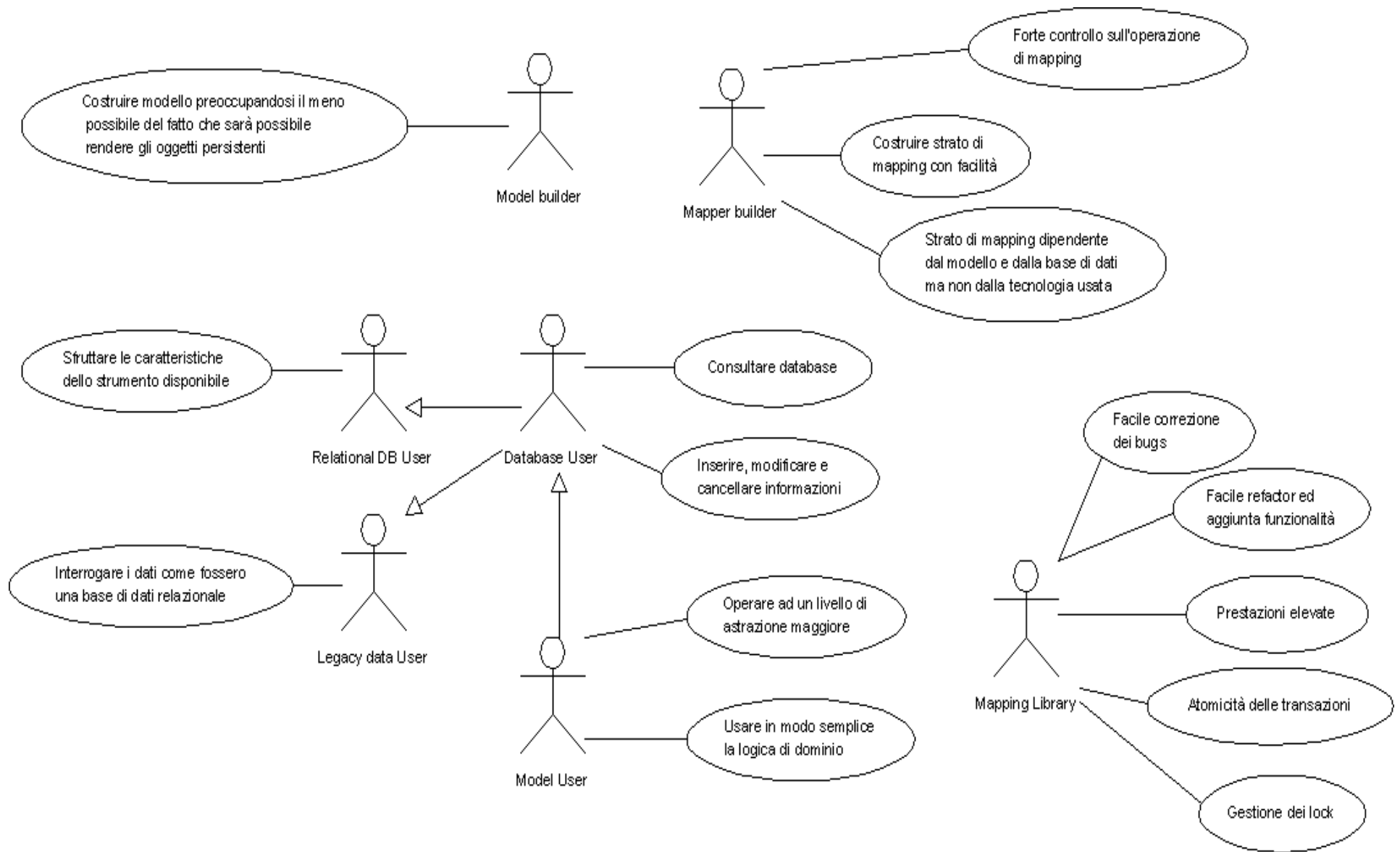
- 3 - Tier

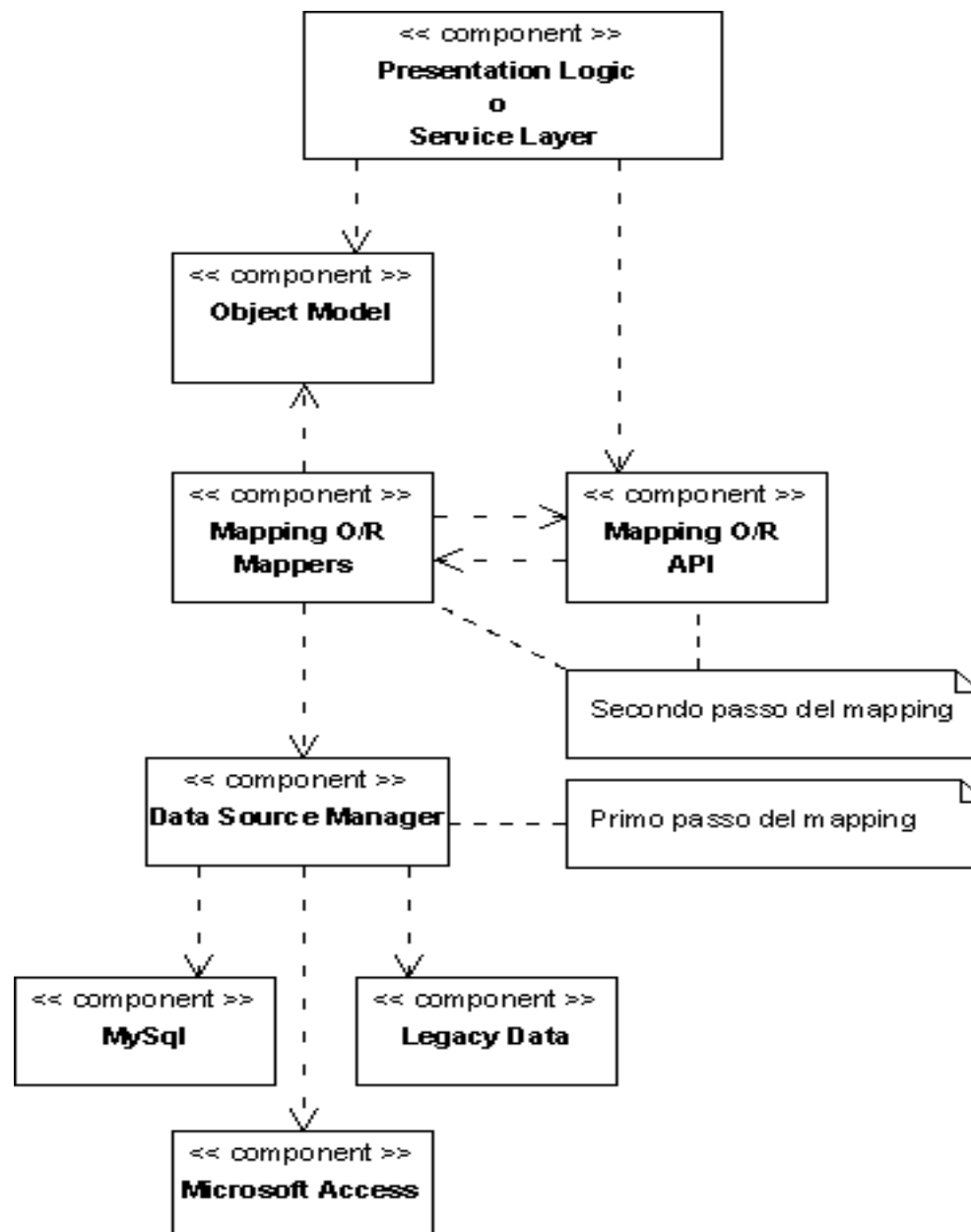


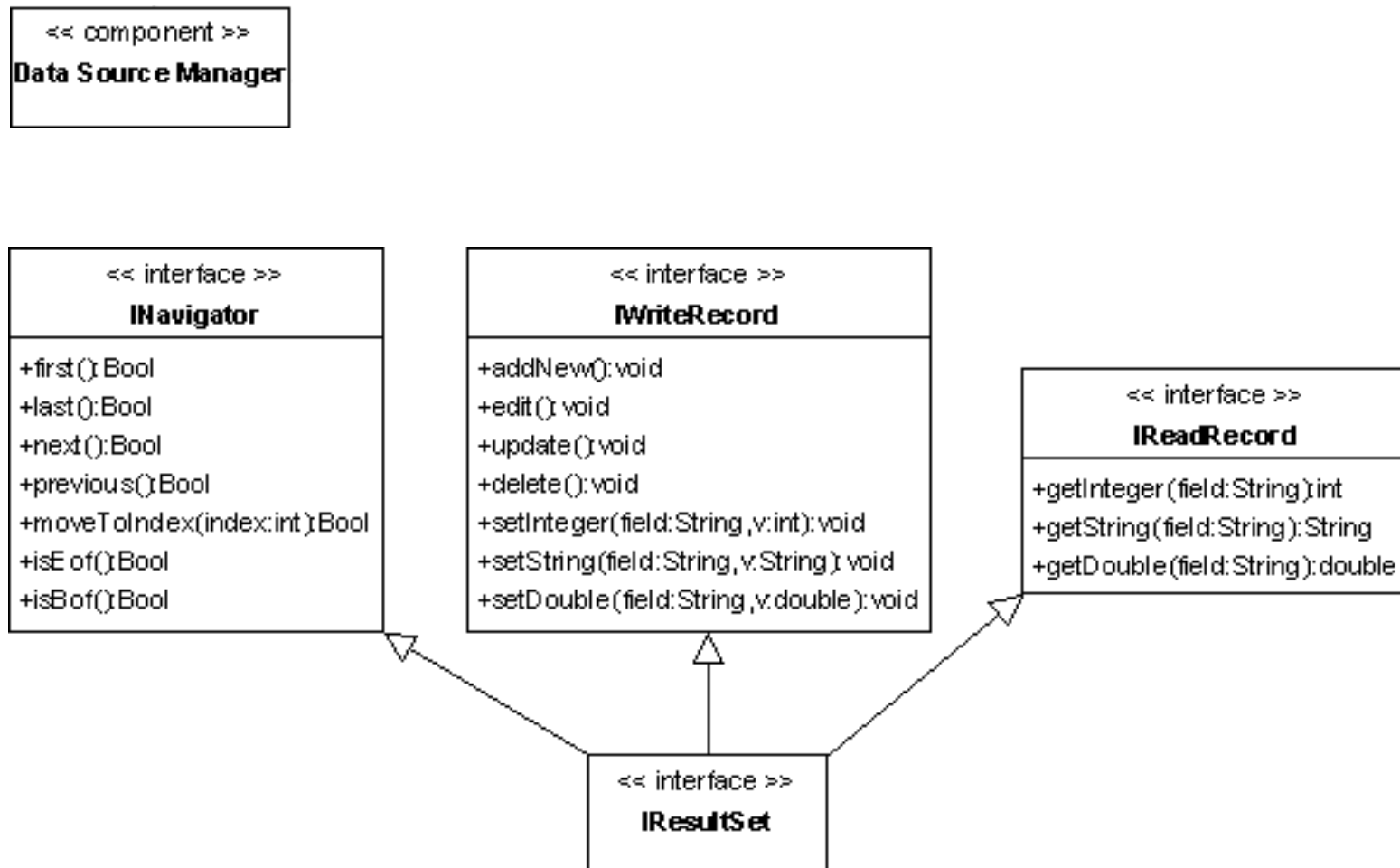
N.B. E' stata adoperata l'architettura 3-Tier

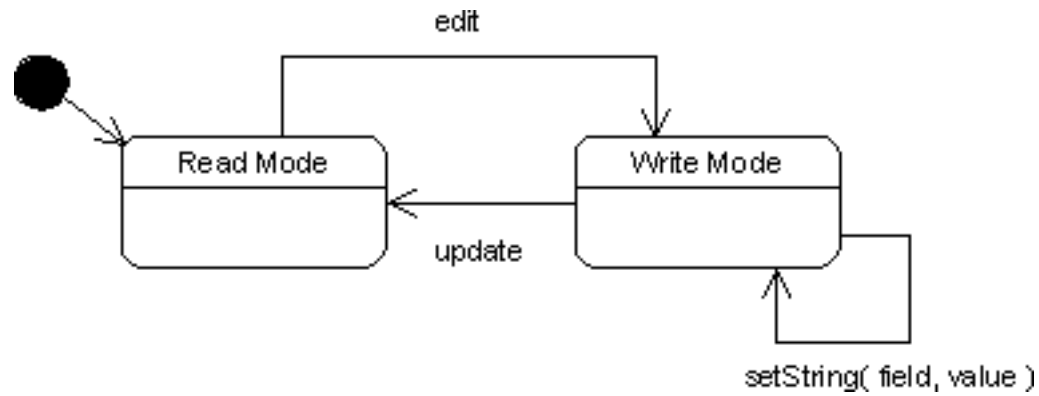
- Modello ad Oggetti per la Logica Applicativa
- Base di dati relazionale per l'archiviazione
- Mapping O/R consiste nell'interfacciare il modello ad oggetti con la base di dati
 - Top down
 - Bottom up
 - Meet in the middle







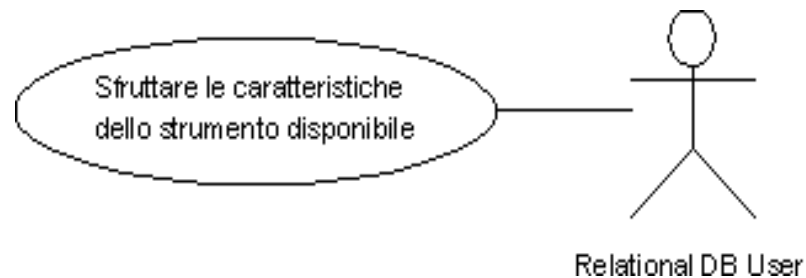




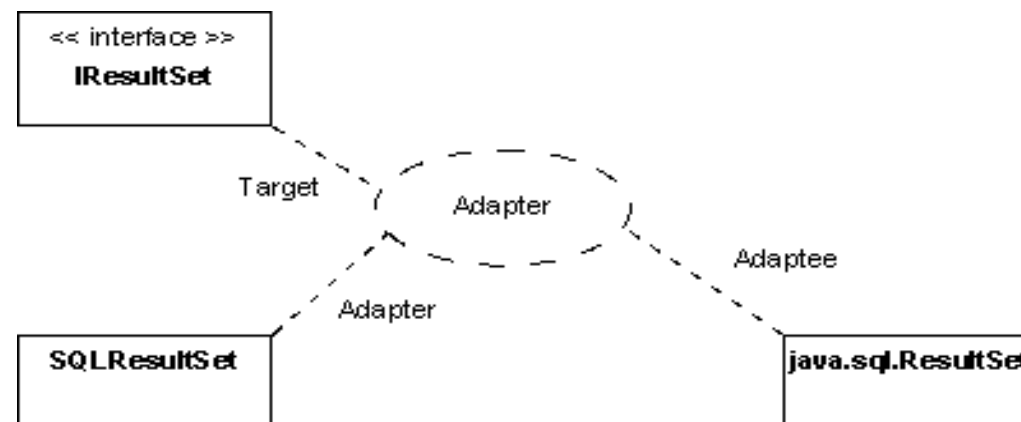
```
/*
 * 1. Leggo il campo 'nome' del primo record di rs
 */
rs.first();
String nome = rs.getString( "nome" );
```

```
/*
 * 2. Modifico il campo cognome
 */
rs.edit();
rs.setString( "cognome", "Rossi" );
rs.update();
```

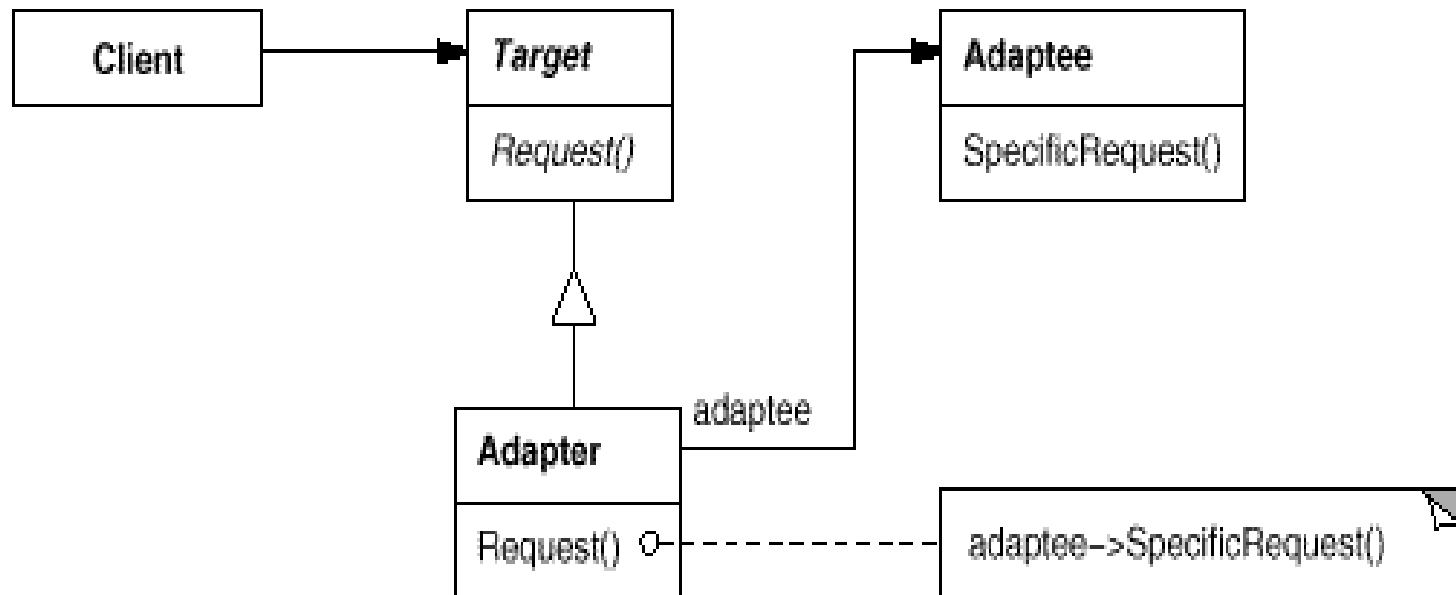
```
/*
 * 3. Mi sposto sul prossimo record e lo cancello
 */
rs.next();
rs.delete();
```



- Design Pattern Adapter
- Utilizzo funzionalità disponibili
- Semplice da realizzare

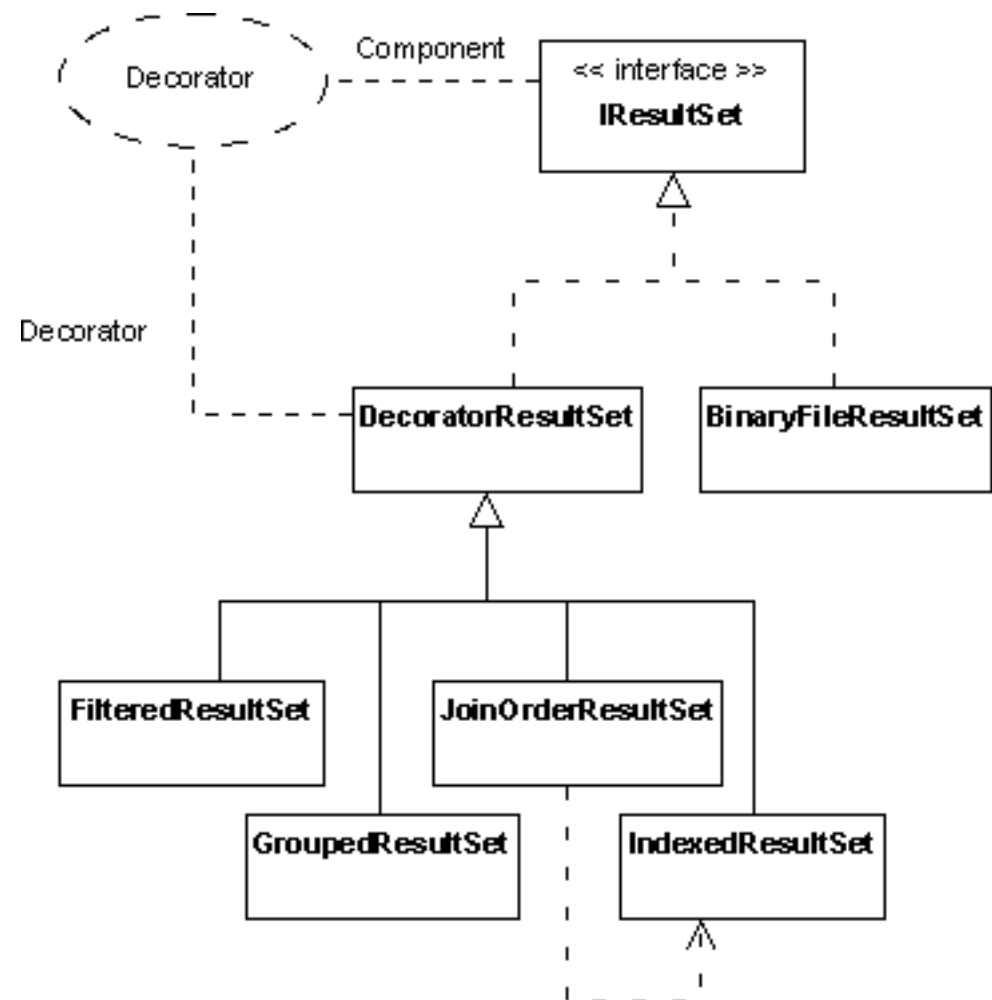


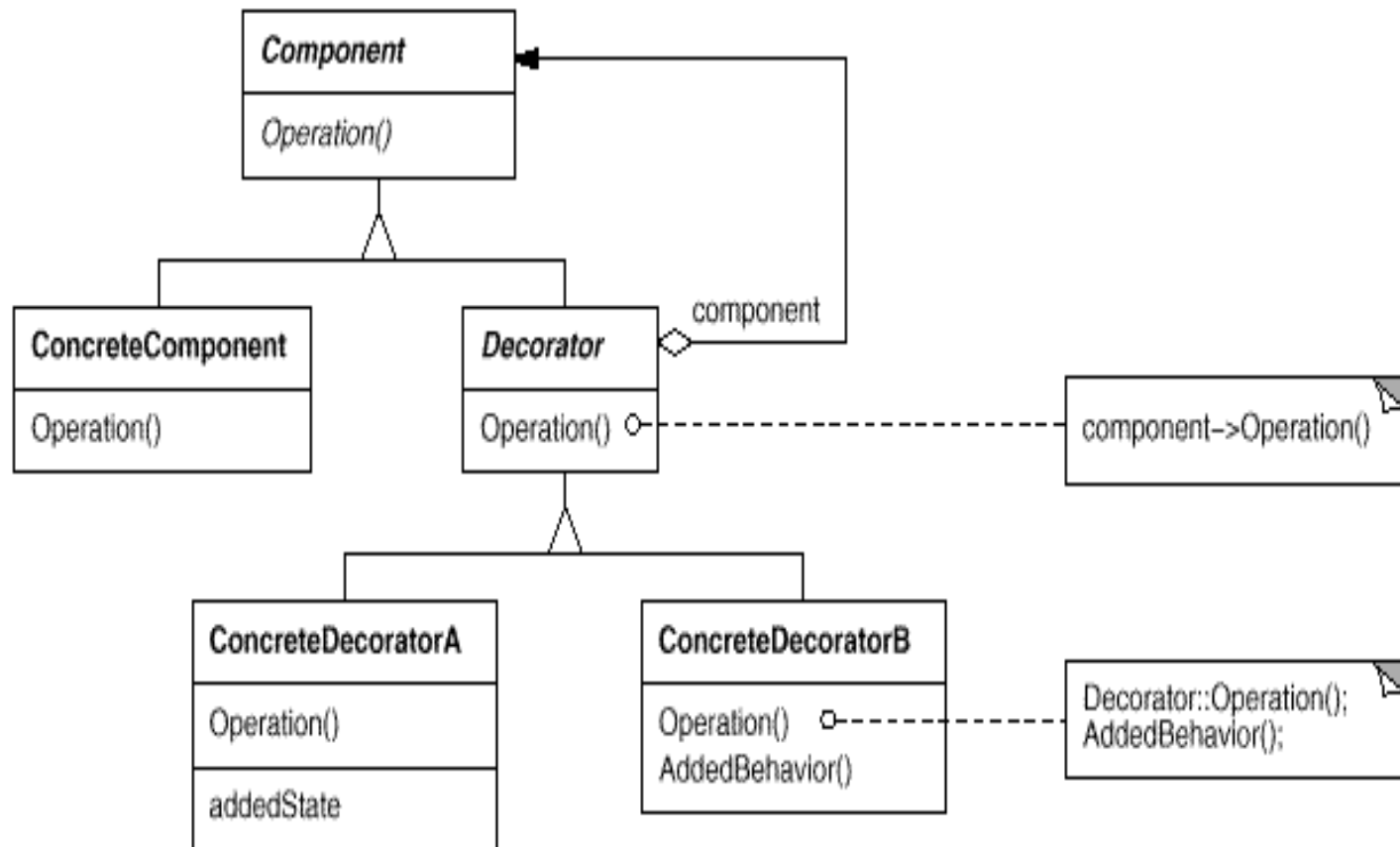
N.B. `ResultSet` e `java.sql.ResultSet` hanno interfacce simili

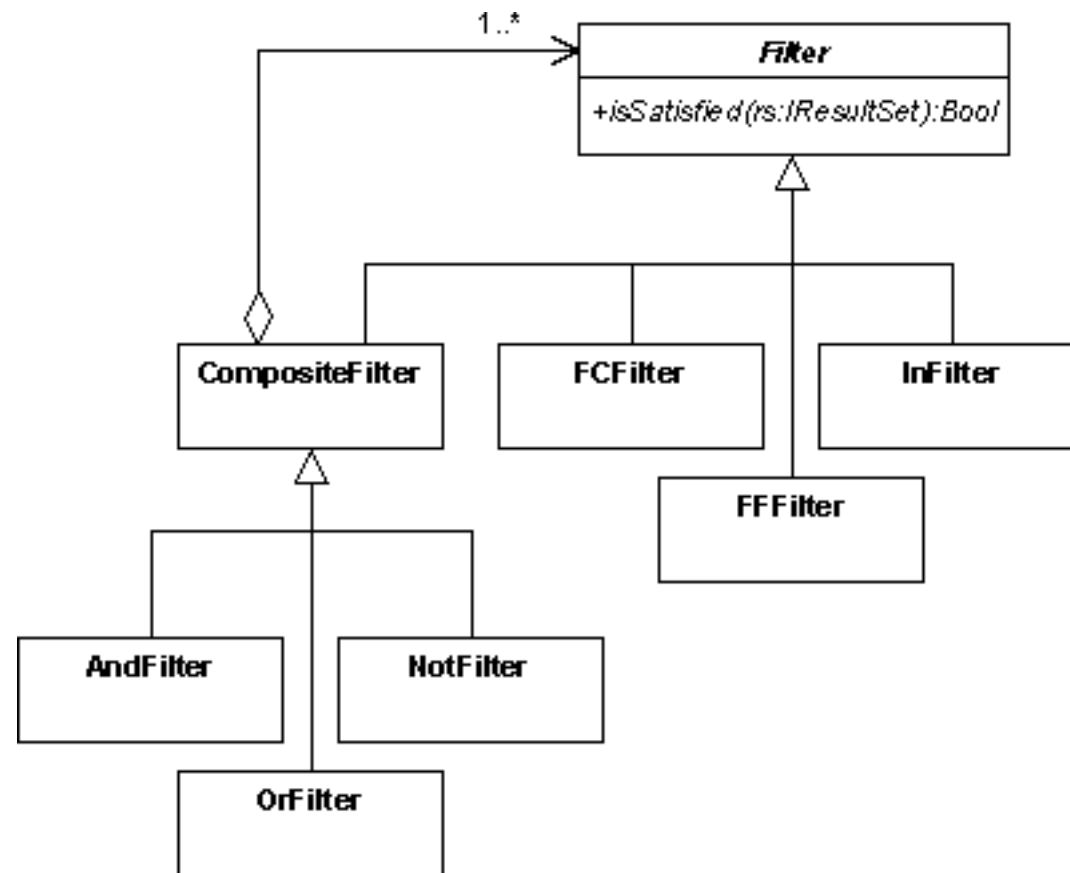


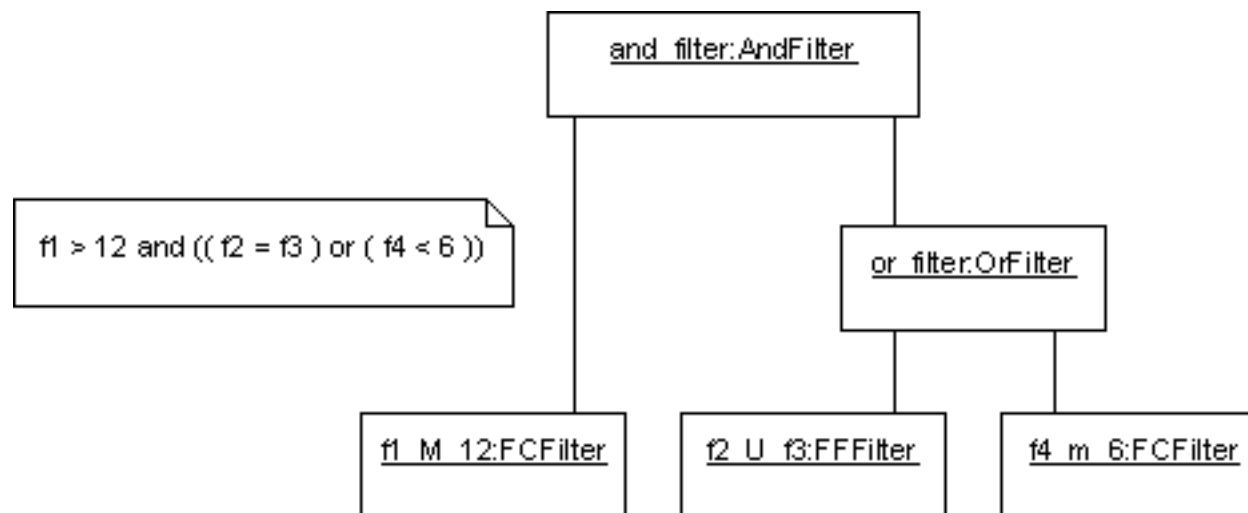


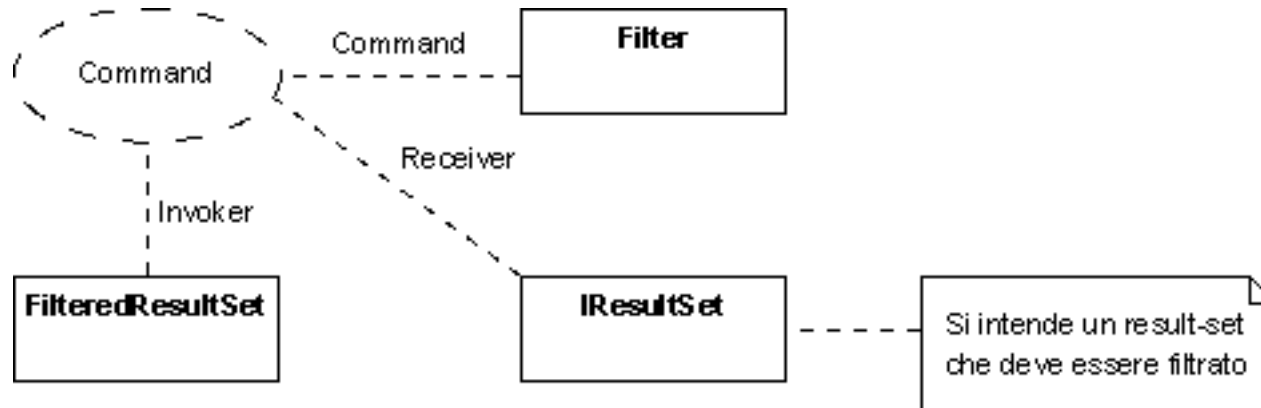
- Design Pattern Decorator
- Si compongono decoratori diversi
- Facile da espandere

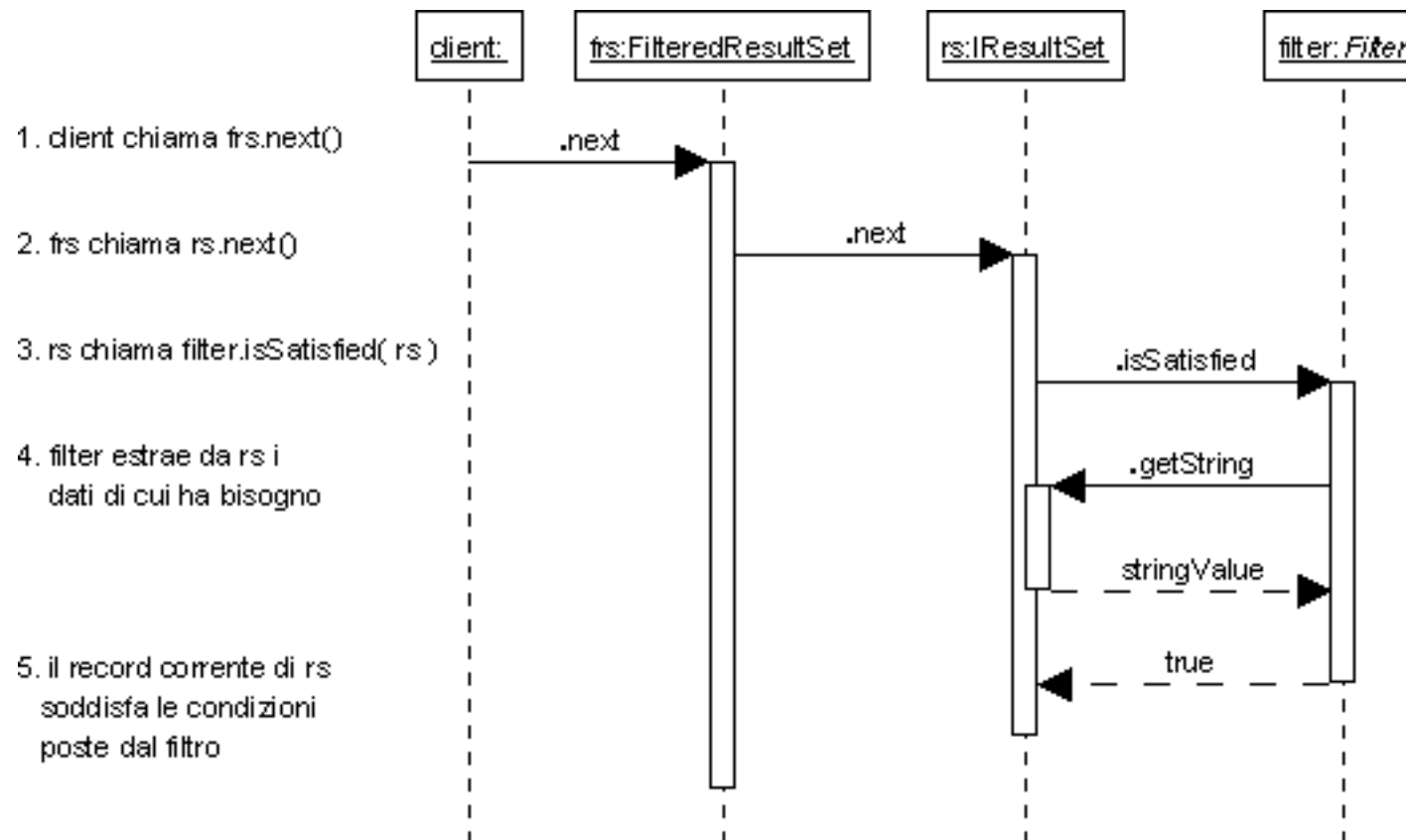


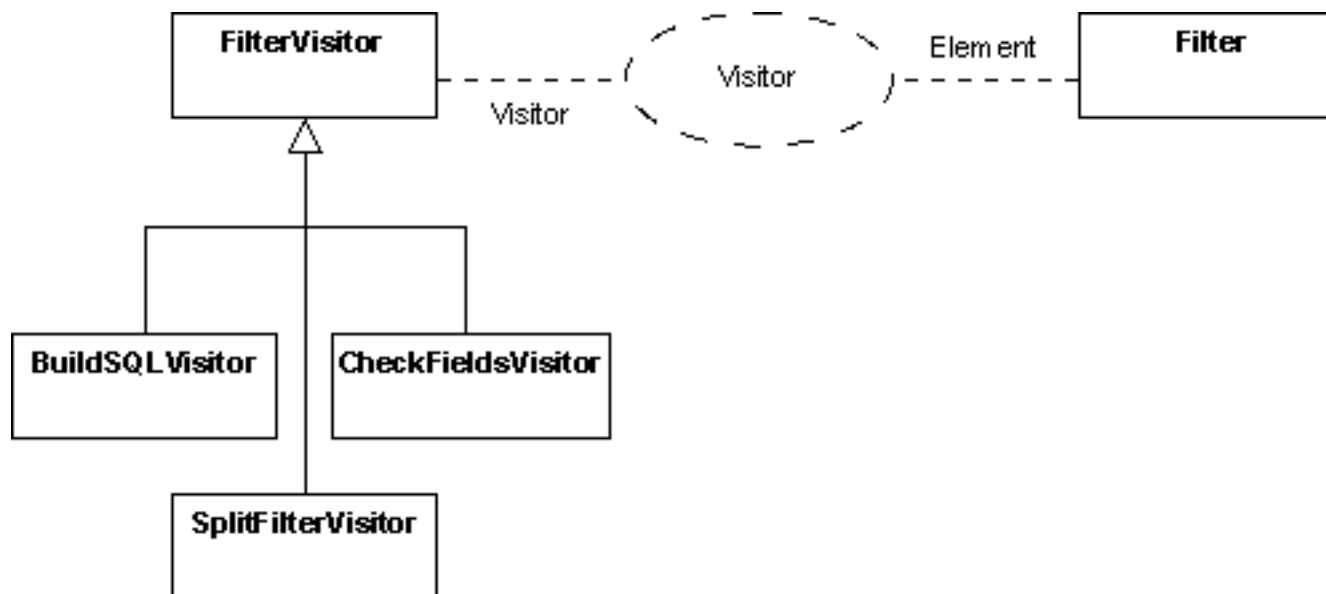


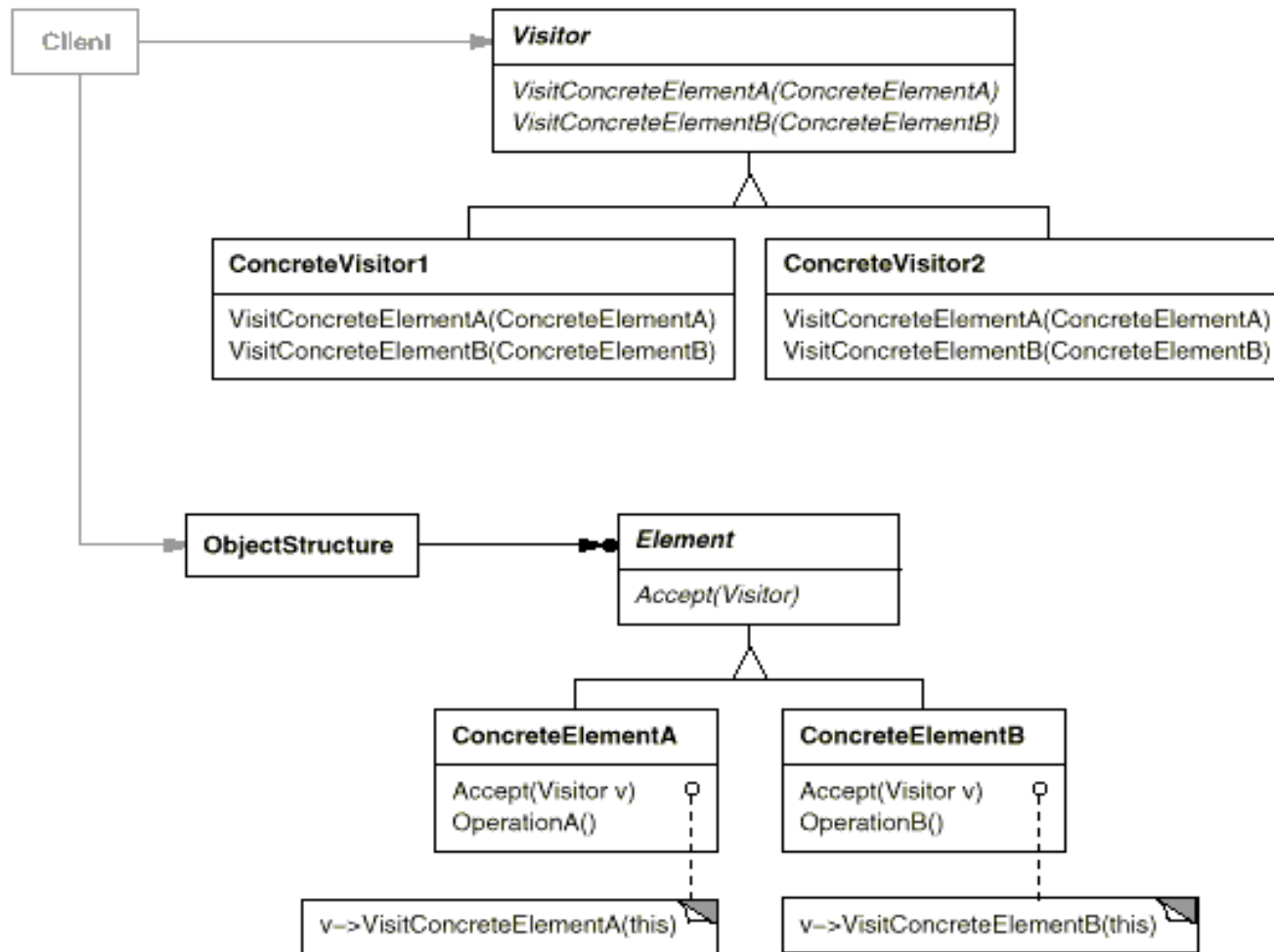


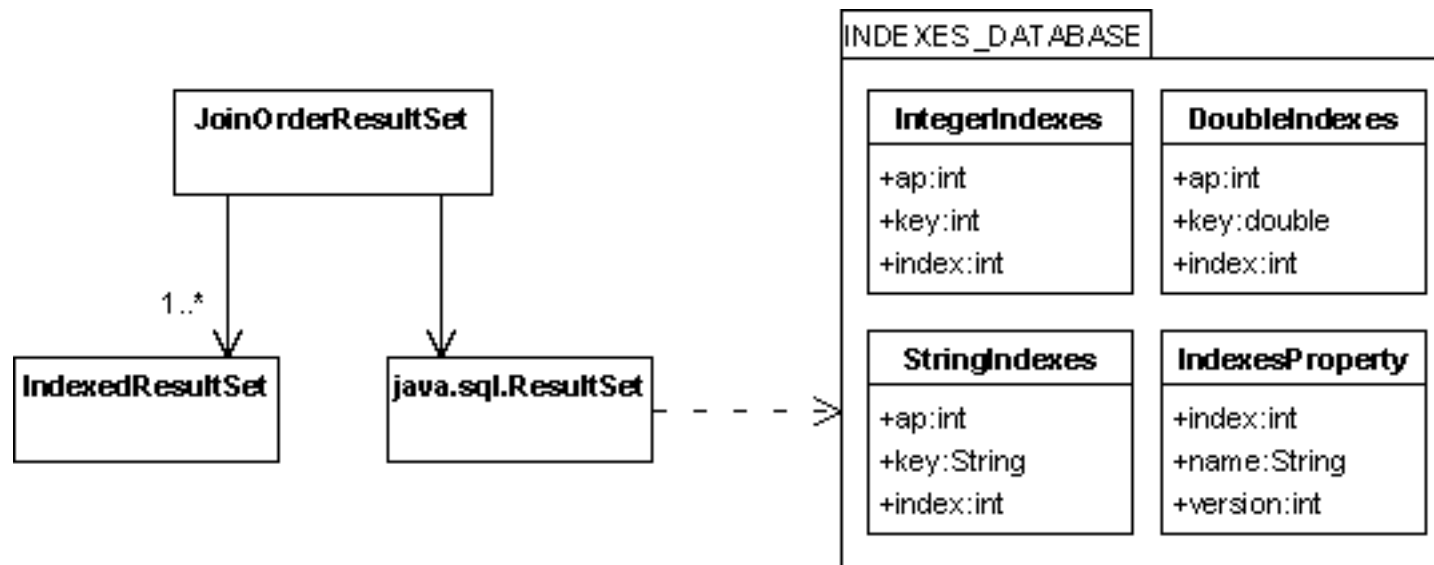


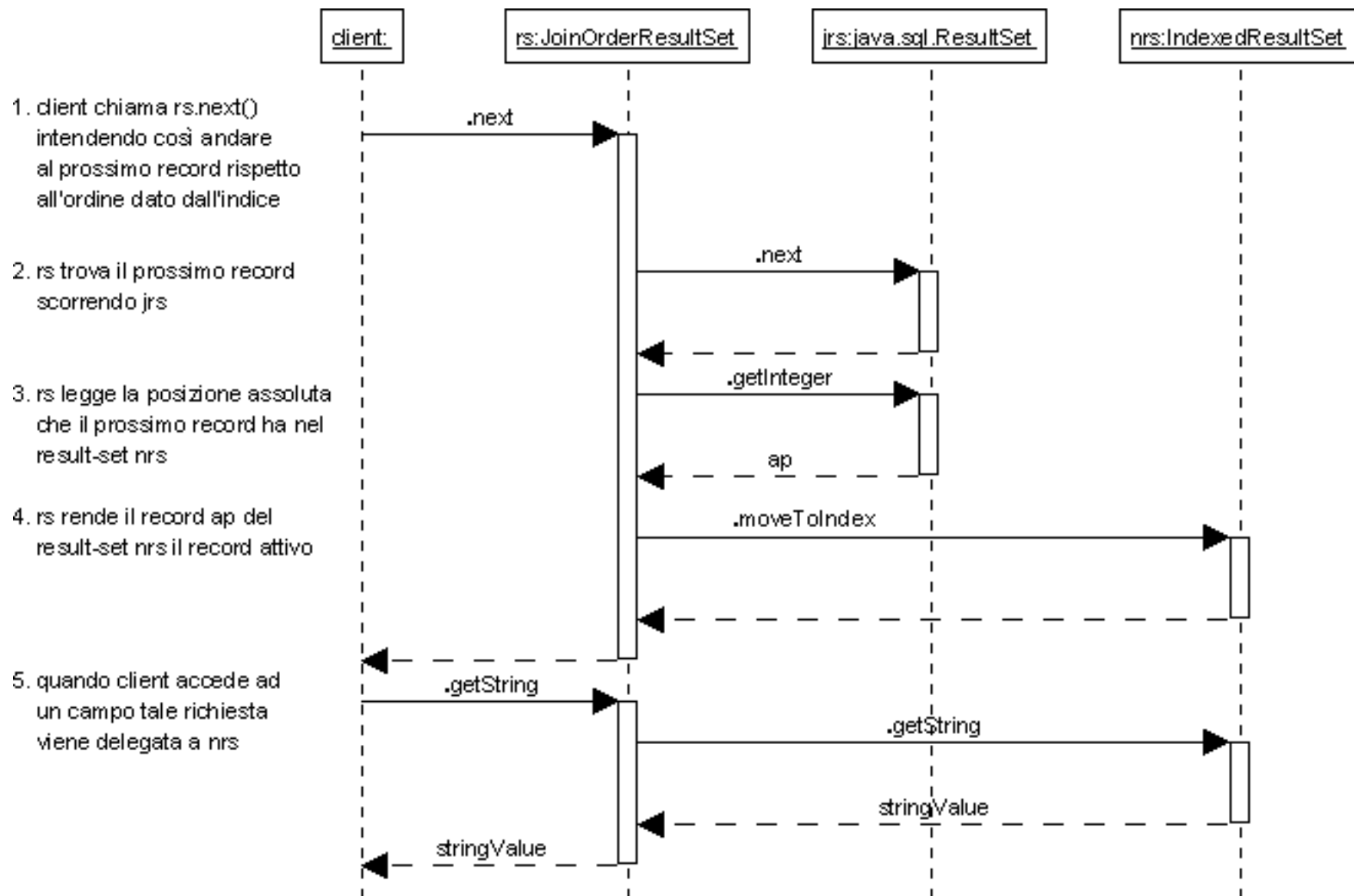












Es.1: selezione delle colonne Cod_Baci e Desc_Baci dalla tabella Bacini

```
from Bacini as Baci
select Baci.Cod_Baci as Codice,
       Baci.Descr_Baci as Descr
order by Codice
```

Es.2: selezione dei bacini il cui codice è maggiore di 10

```
from Bacini as Baci
select Baci.Cod_Baci as Codice,
       Baci.Descr_Baci as Descr
where Codice > 10
order by Codice
```

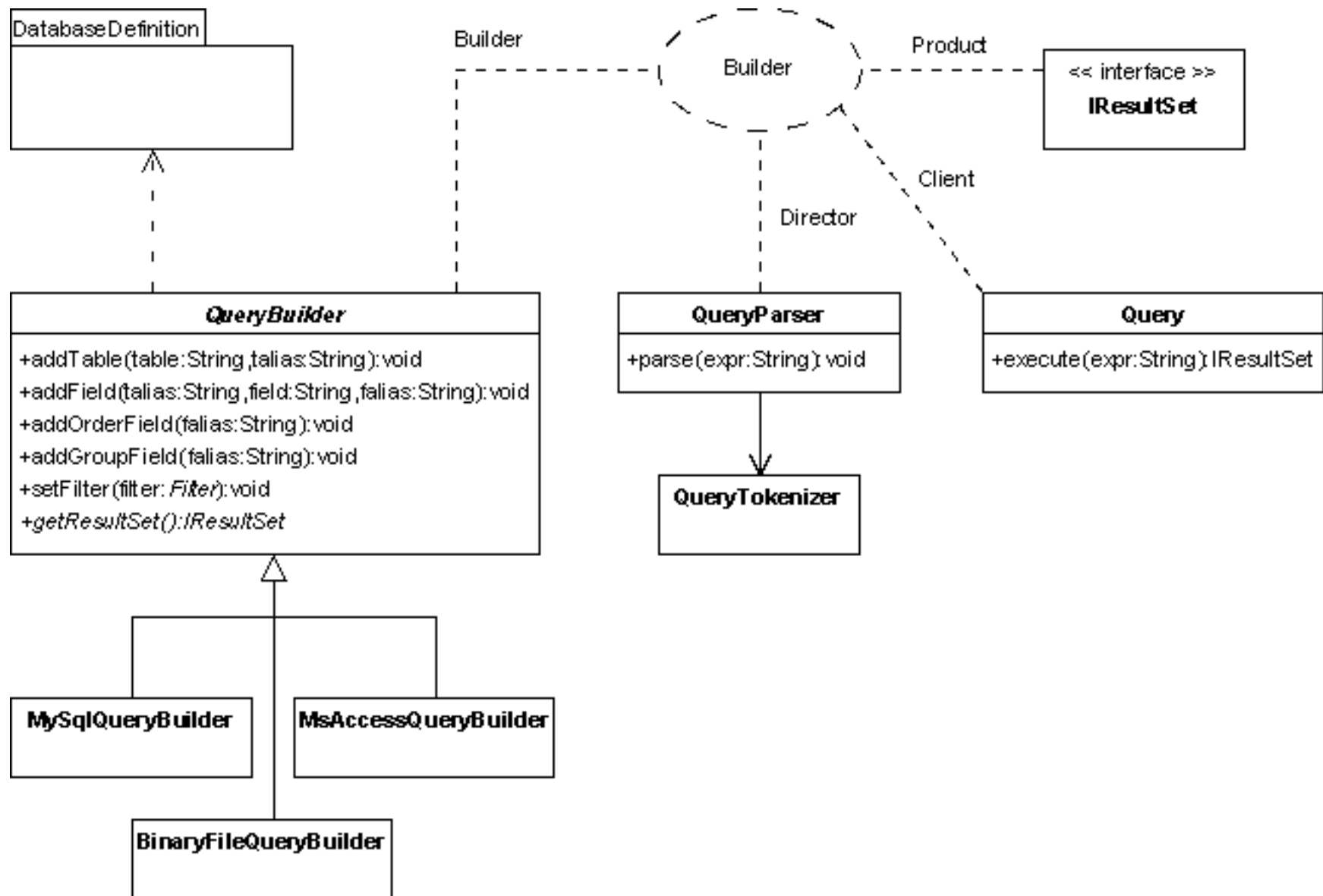
Es.3: selezione delle stazioni con la descrizione del bacino cui appartengono

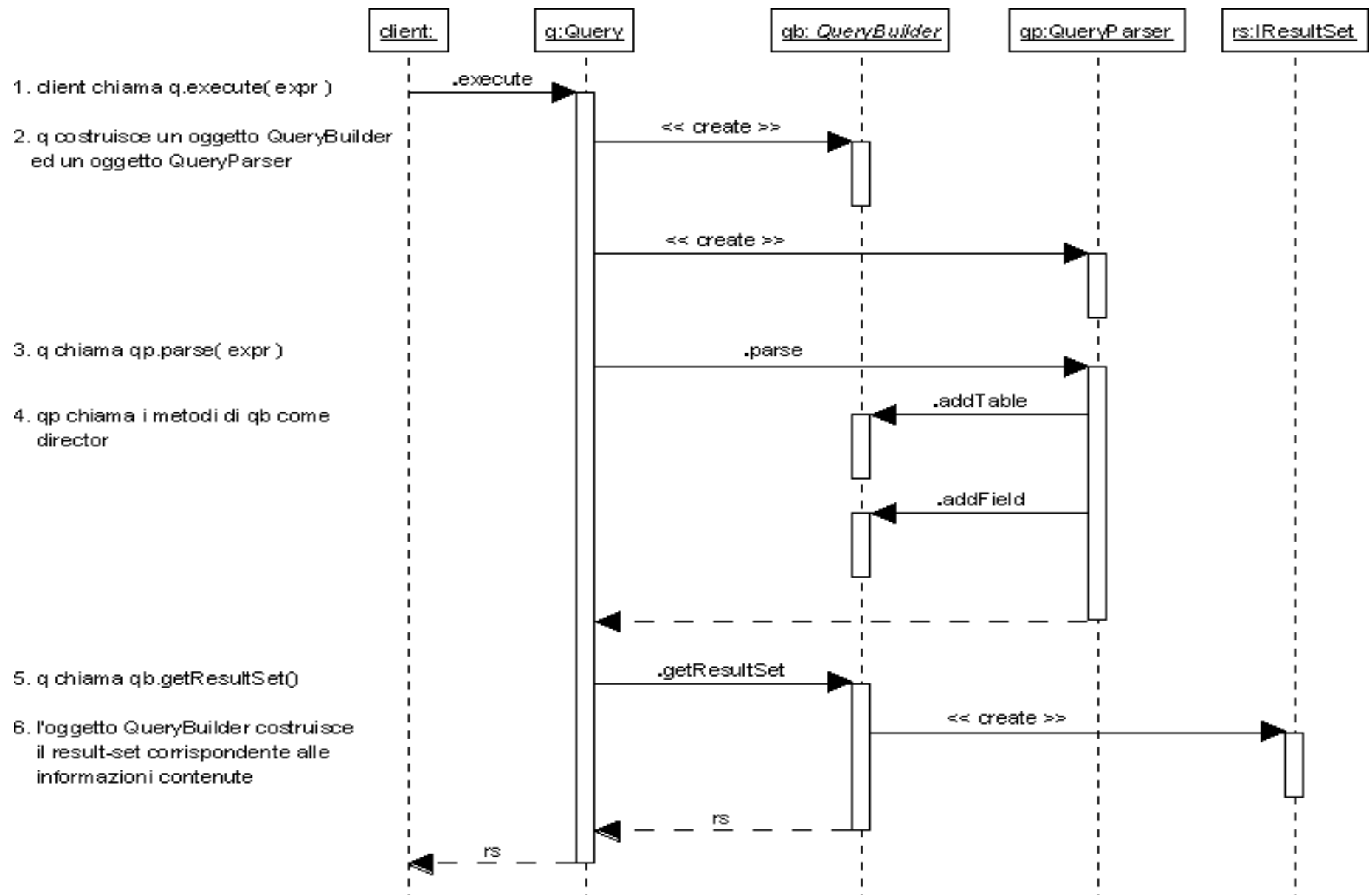
```
from Bacini as Baci, Stazioni as Staz
select Baci.Descr_Baci as DBaci,
       Staz.Descr_Staz as DStaz,
       Baci.Id_Baci,
       Staz.RifBaci_Staz
where RifBaci_Staz = Id_Baci
```

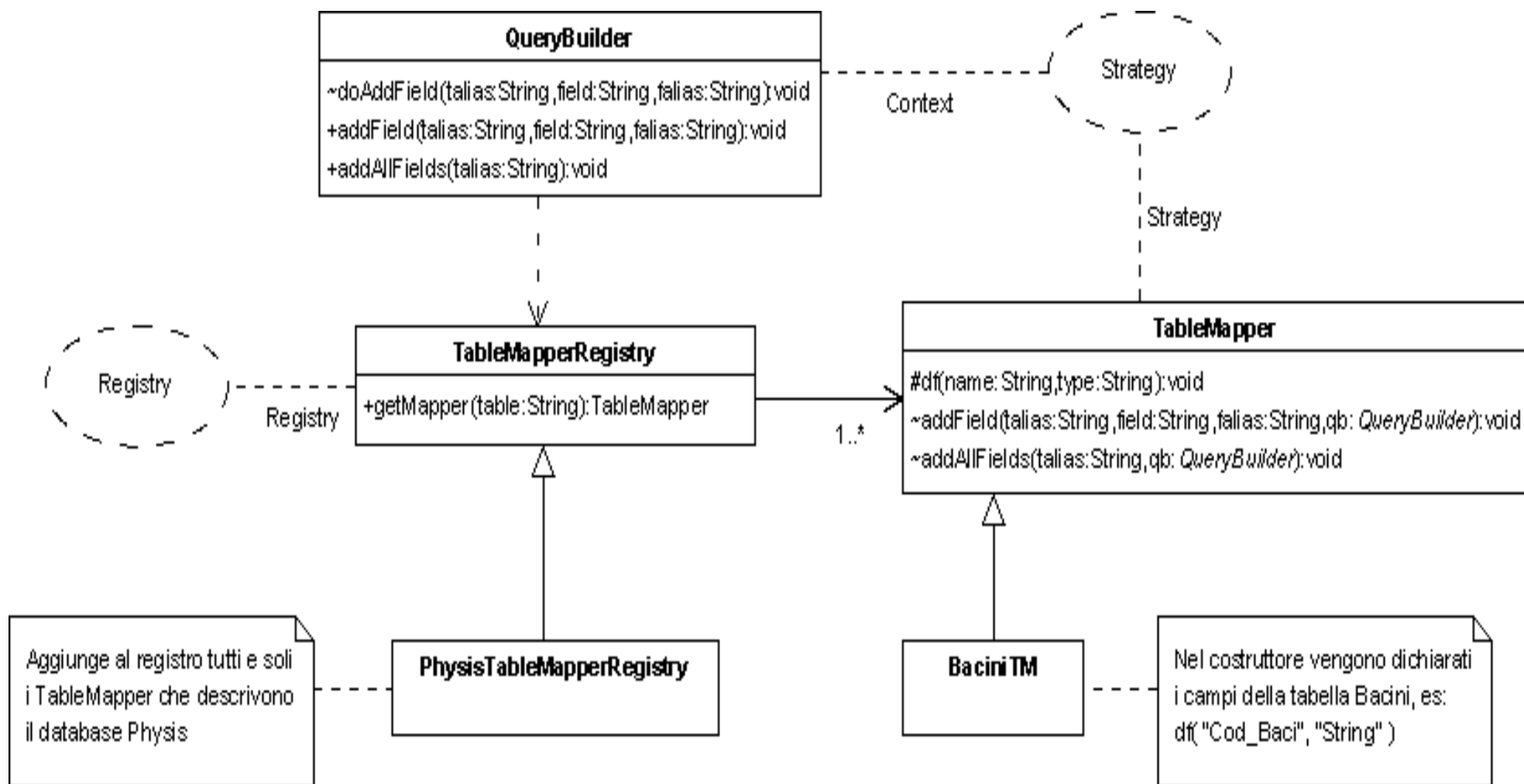
Es.4: selezione dei bacini di un gruppo

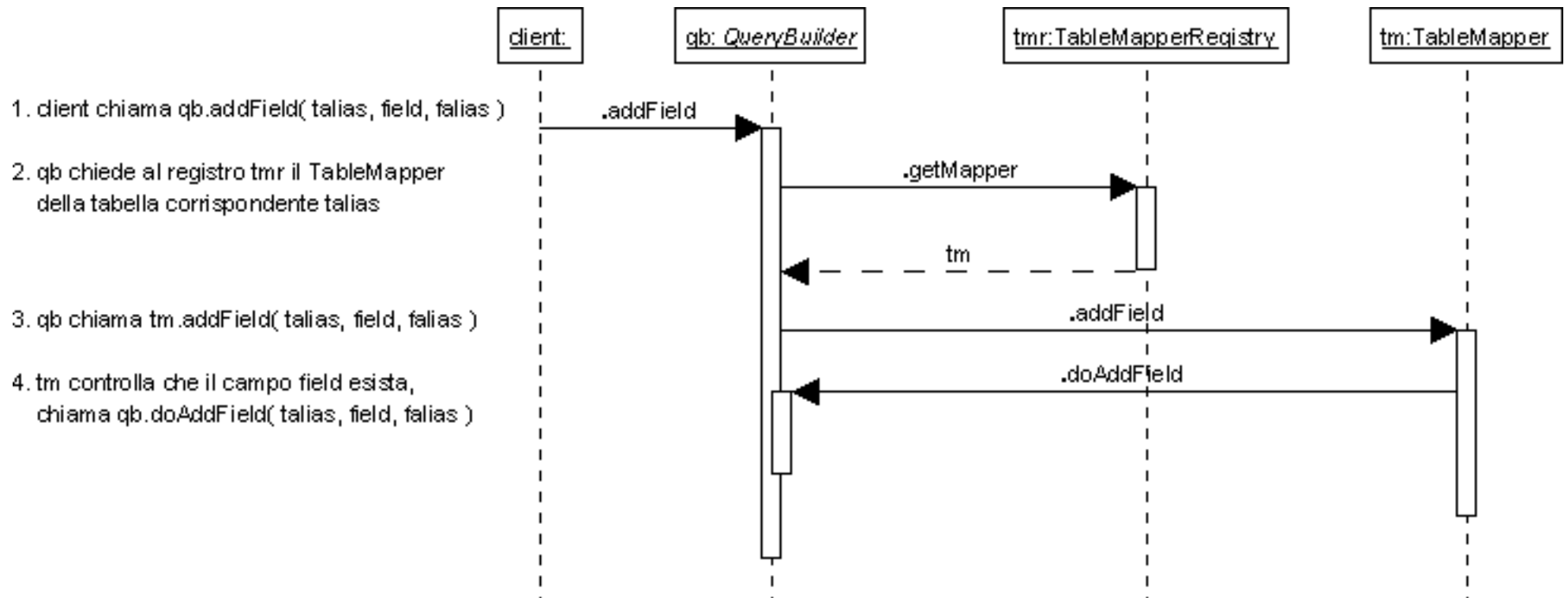
```
from Bacini as Baci
select Baci.*
where Id_Baci in ( [10], [20], [30] )
```

- 1) <mysql> -> <from> <select> <where> <groupBy> <orderBy>
- 2) <from> -> **from** <tablesList>
- 3) <select> -> **select** <fieldsList>
- 4) <where> -> **where** <expr> | ϵ
- 5) <groupBy> -> **group by** <fieldsAliasList> | ϵ
- 6) <orderBy> -> **order by** <fieldsAliasList> | ϵ
- 7) <tablesList> -> <table> <tablesListAgain>
- 8) <tablesListAgain> -> , <tablesList> | ϵ
- 9) <table> -> **identifier** <tableAlias>
- 10) <tableAlias> -> **as alias** | ϵ
- 11) <fieldsList> -> <field> <fieldsListAgain>
- 12) <fieldsListAgain> -> , <fieldsList> | ϵ
- 13) <field> -> **identifier.identifier** <fieldAlias> | **identifier.***
- 14) <fieldAlias> -> ->**as identifier** | ϵ
- 15) <fieldsAliasList> -> **identifier** <fieldsAliasListAgain>
- 16) <fieldsAliasListAgain> -> , <fieldsAliasList> | ϵ
- 17) <exprLp> -> <exprHp> <exprLpAgain>
- 18) <exprHp> -> <expr> <exprHpAgain>
- 19) <exprLpAgain> -> **or** <exprLp> | ϵ
- 20) <expr> -> (<exprLp>) | **not** (<exprLp>) | <termExpr>
- 21) <exprHpAgain> -> **and** <exprHp> | ϵ
- 22) <termExpr> -> **identifier** <termRigthExpr>
- 23) <termRightExpr> -> (< | = | >) (**identifier** | **const**) | **in** (<keysList>) | **is key**
- 24) <keysList> -> **key** <keysListAgain>
- 25) <keysListAgain> -> , <keysList> | ϵ

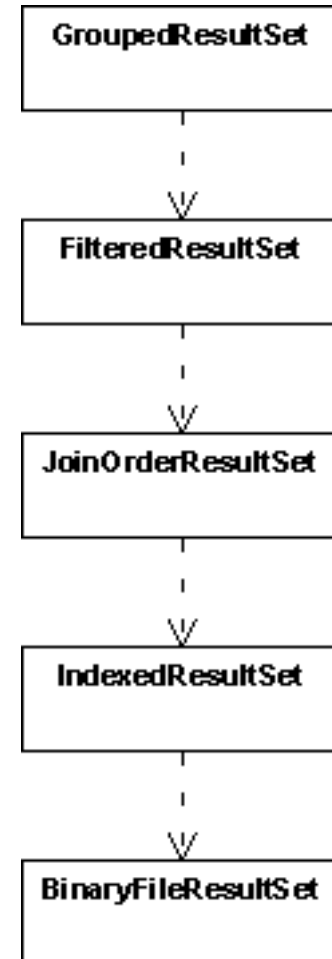


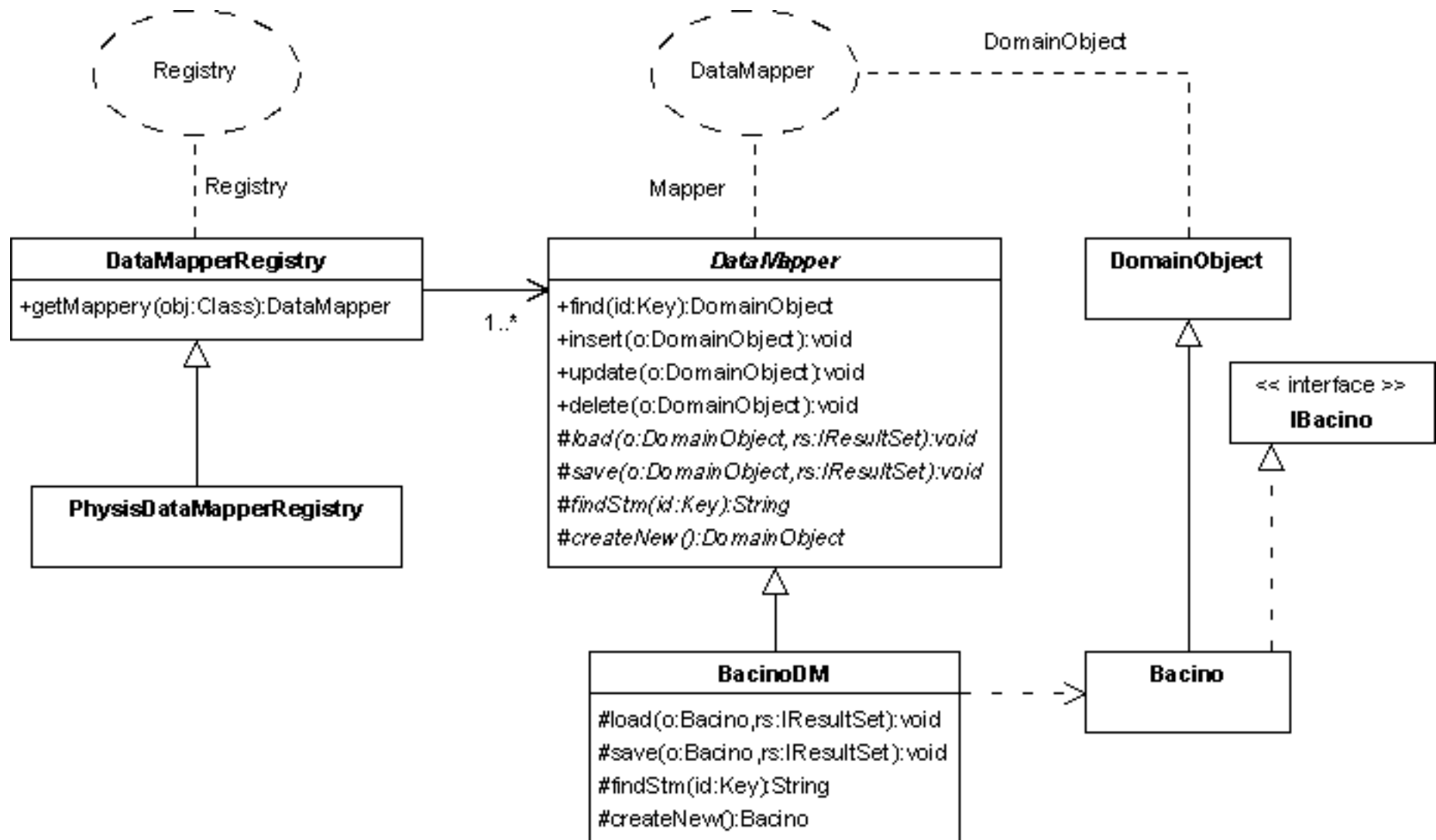


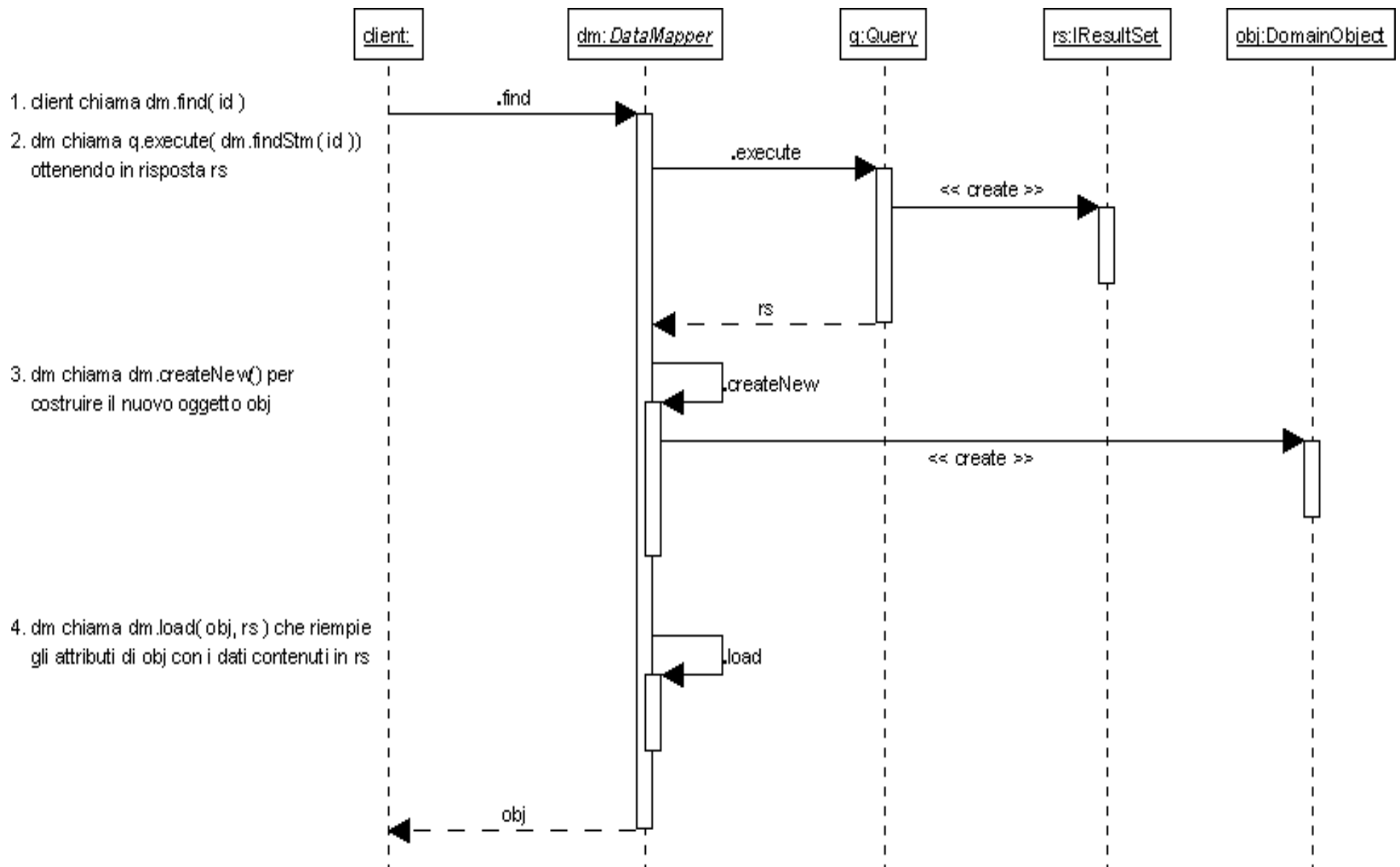


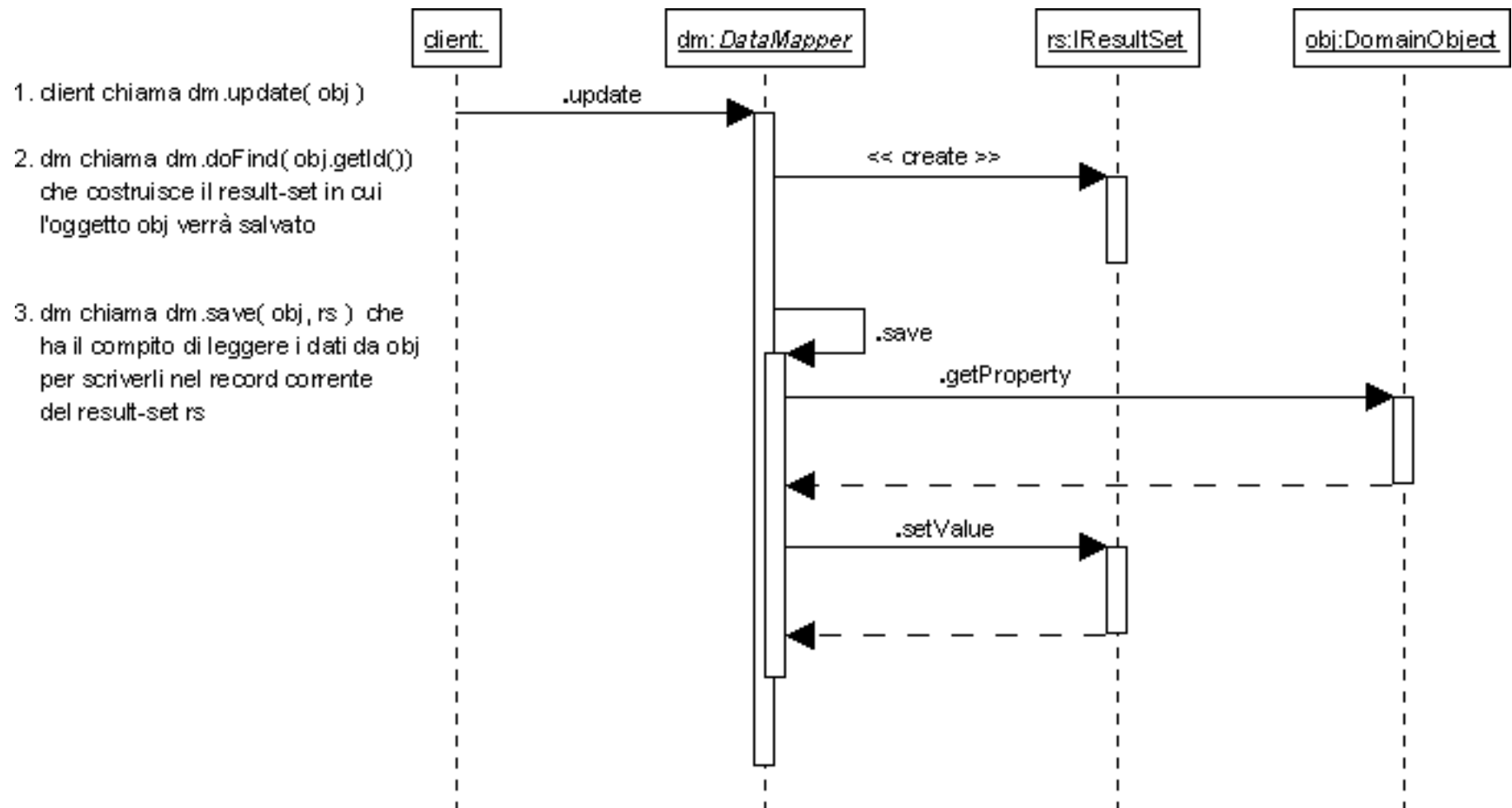


- **Database Relazionale**
 - QueryBuilder costruisce un'espressione SQL propria dello strumento DBR in uso
 - La query ottenuta viene eseguita per mezzo di JDBC
 - Il resultset di tipo `java.sql.ResultSet` viene adattato all'interfaccia `IResultSet`
- **Legacy Data**
 - QueryBuilder costruisce il result-set di risultato come composizione attraverso l'uso di opportuni decoratori di uno o più result-set di base









```
public class GrandezzaDM extends DataMapper<IGrandezza> {
    public GrandezzaDM( ISession session ) {
        super( session );
    }

    public IGrandezza createNew() {
        IGrandezza o = new Grandezza();
        setPriority( o, 5 );
        setSession( o );
        markNew( o );
        return o;
    }

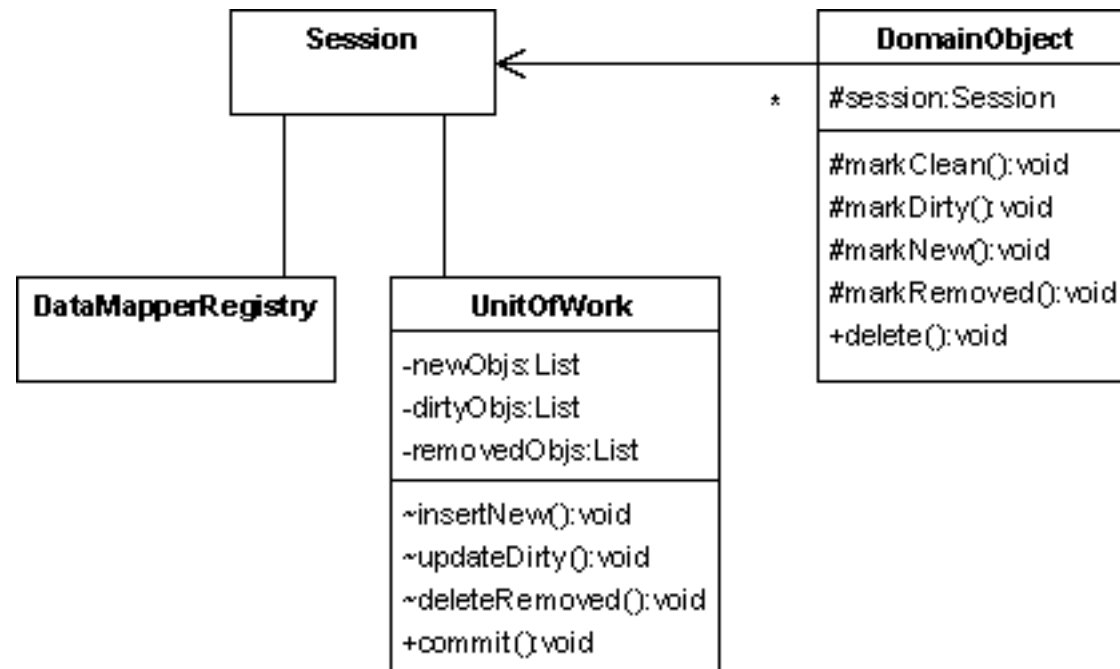
    protected void load(IFieldValue id, IResultSet rs, IGrandezza o) throws MappingException {
        o.setCodice( rs.getInteger( "Cod_Gran" ) );
        o.setDescrizione( rs.getString( "Descr_Gran" ) );
        o.setUnitàMisura( rs.getString( "UdM_Gran" ) );
        o.setSigla( rs.getString( "Sigla_Gran" ) );
        o.setParte( rs.getInteger( "Parte_Gran" ) );
    }

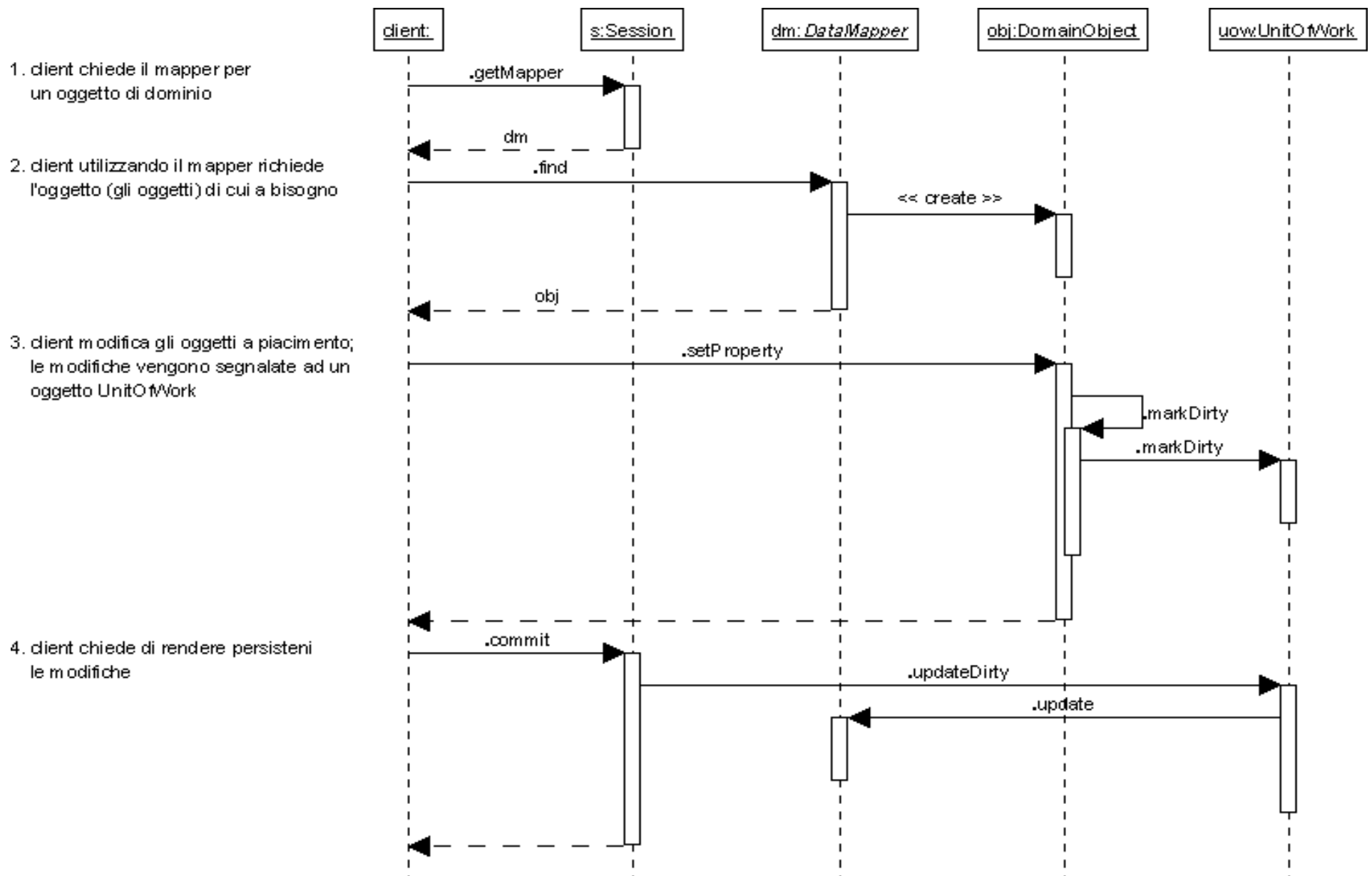
    protected void save(IResultSet rs, IGrandezza o) throws MappingException {
        rs.setInteger( "Cod_Gran", o.getCodice());
        rs.setString( "Descr_Gran", o.getDescrizione());
        rs.setString( "UdM_Gran", o.getUnitàMisura());
        rs.setString( "Sigla_Gran", o.getSigla());
        rs.setInteger( "Parte_Gran", o.getParte() );
    }

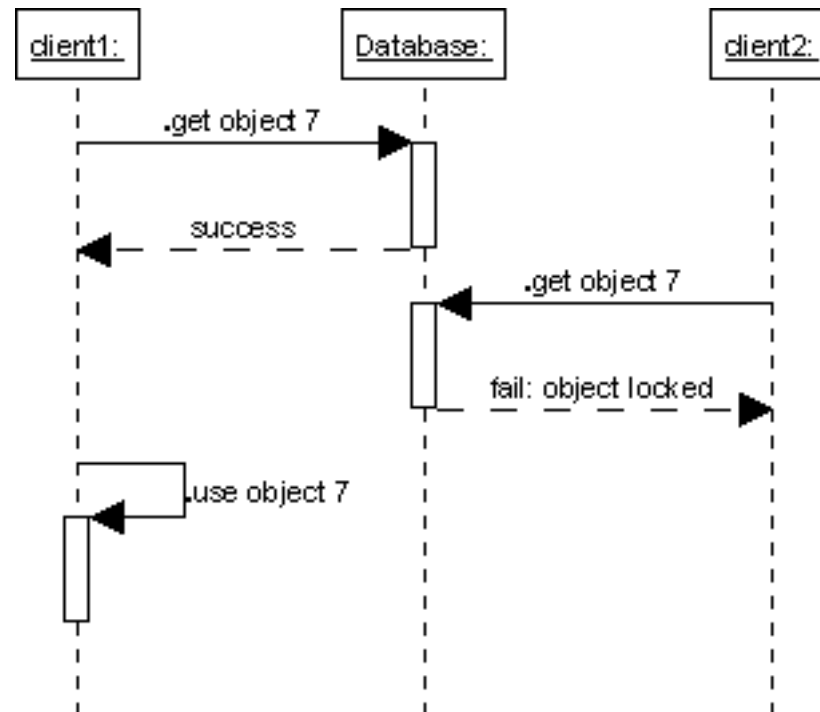
    protected FindStatement findByldStm( IFieldValue id ) {
        return new FindStatement( "Id_Gran",
            "from Grandezze as gr " +
            " select gr.* " +
            " where Id_Gran is " + id.toKey());
    }
}
```

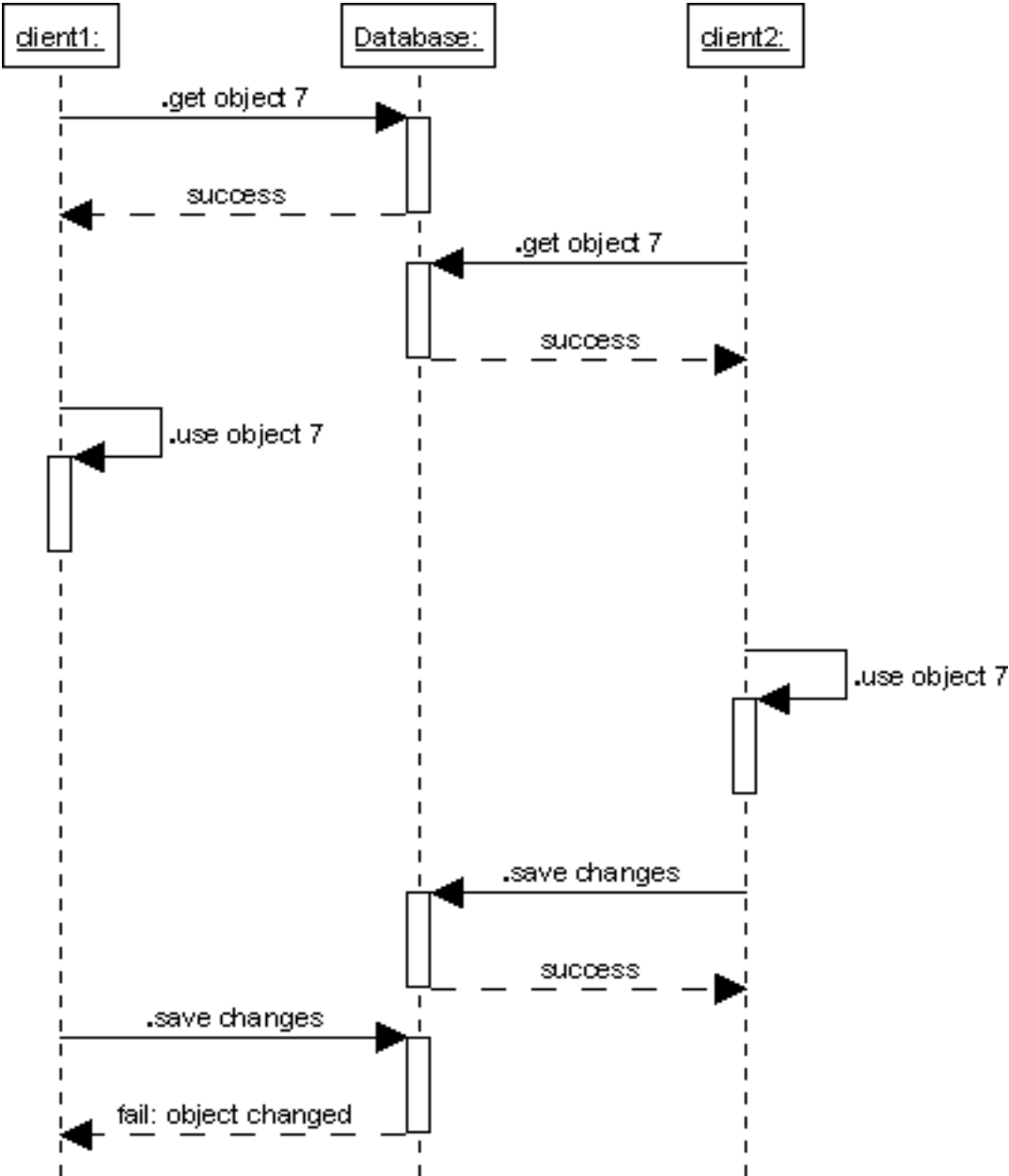

- **Lettura**
 - Metodi Finder
 - Cache
 - Proxy
 - Lazy Load

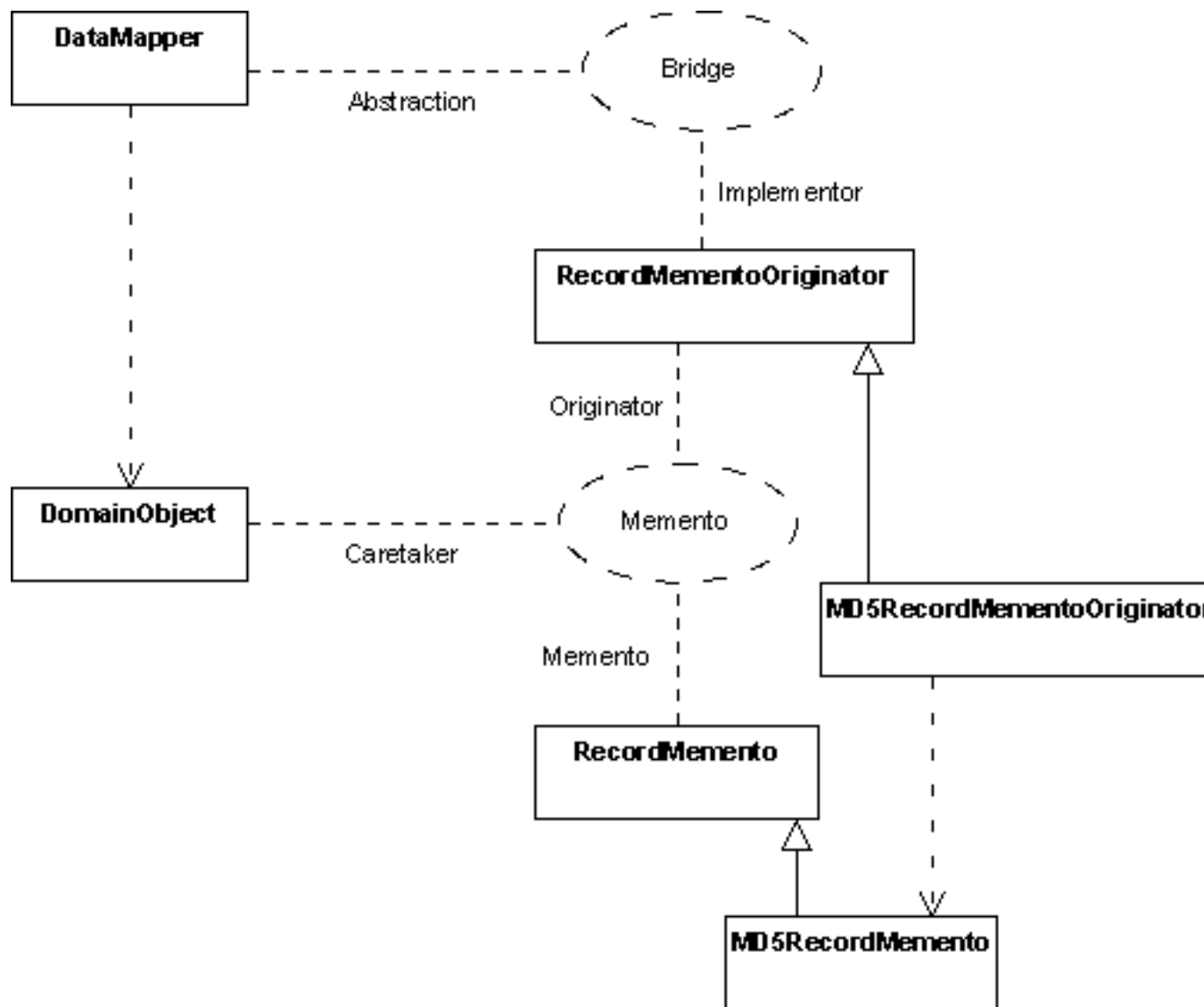
- **Scrittura**
 - Integrità referenziale ed ordine di aggiornamento
 - Business Transaction
 - Concorrenza

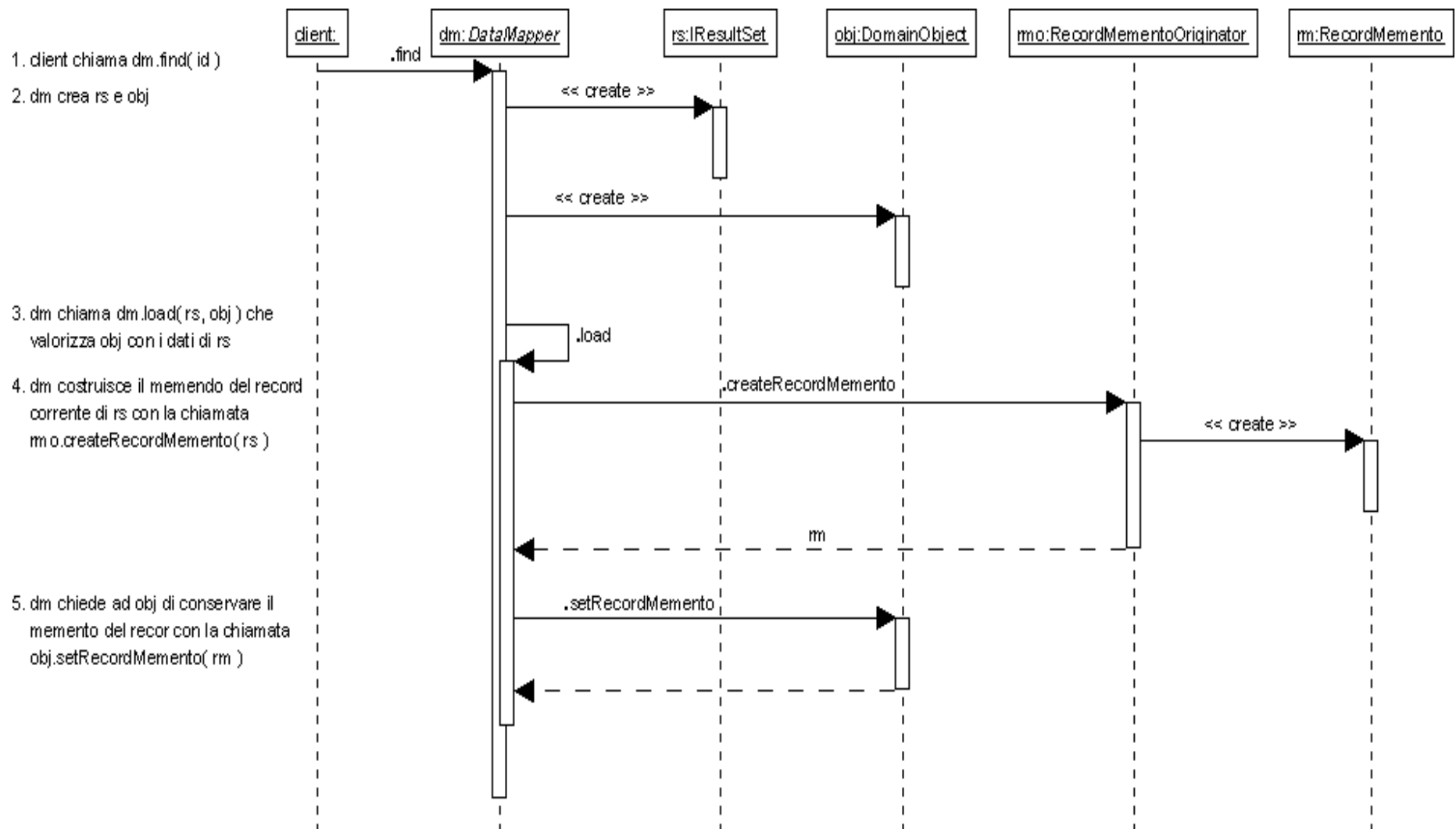


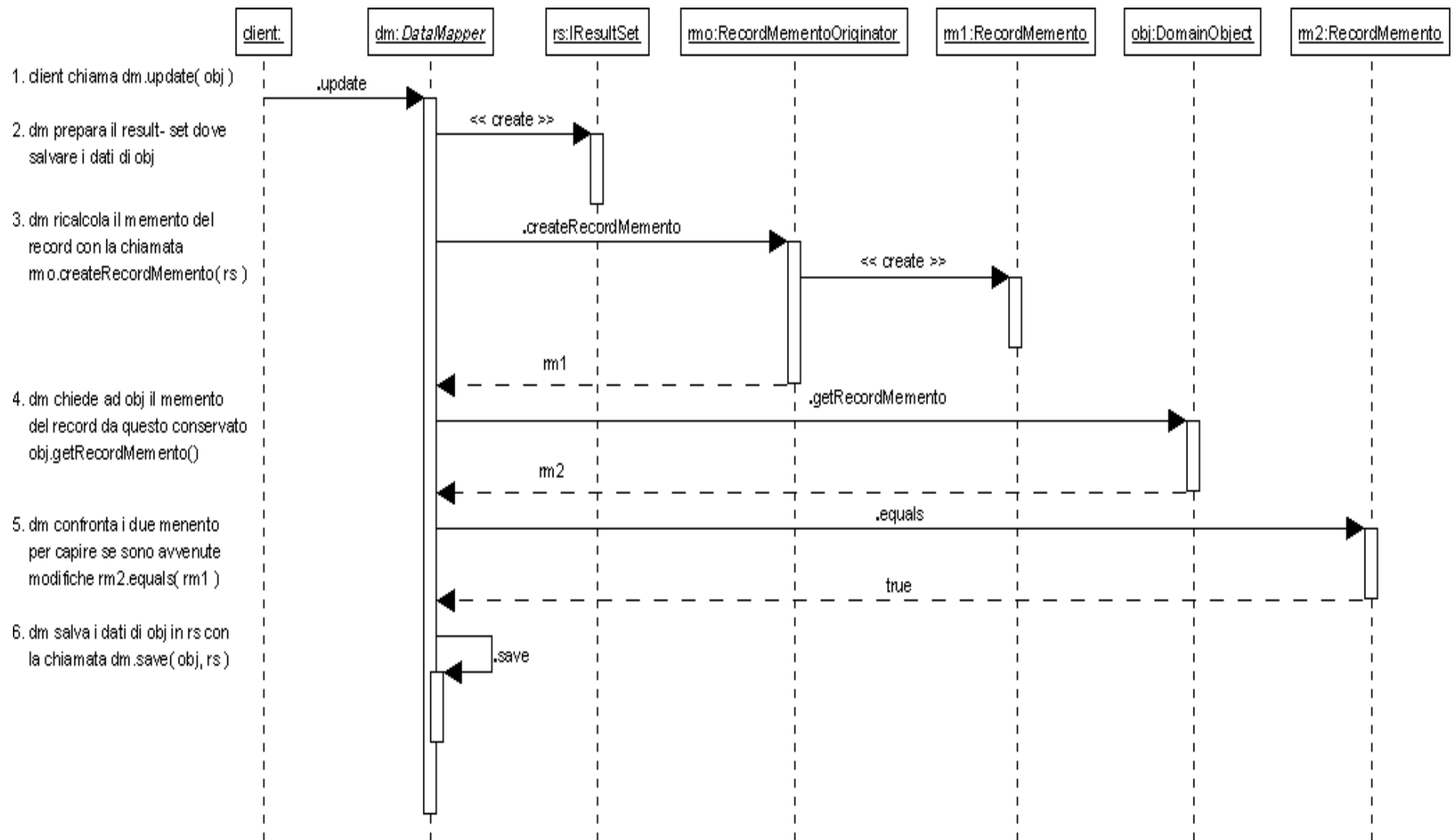












- **Risultati**
 - Strumento per il Mapping
 - Linguaggio di interrogazione indipendente dalla sorgente

- **Sviluppi possibili**
 - Aumentare espressività linguaggio
 - Metadata mapping
 - Supporto di modelli di dominio arbitrari