If the designers of query languages have worked hard to make them readable, why not let developers include verbatim LINQ query snippets in Java?

Advantages of Lightweight Language Nesting:

- the original syntax of LINQ is supported without extending the Java grammar
- far less complex than bytecode manipulation. In fact, not even necessary to build AST nodes yourself: just let the parser run again
- the same error reporting mechanism of the Java compiler is re-used for ill-formed LINQ queries
- applicable to other query languages (OQL, XQuery, JPQL, SQL)
Now that we are talking about compiler plugins.

Did you know that the Scala compiler supports a richer extension model than its Java counterparts?

Essentially the same kind of AST transformations:

- LINQ: Inline query continuation
  ```
  from x1 in e1 ... into x2 ...
  -- from x2 in (from x1 in e1 ...) ...
  ```

- SQO: From followed by a join with into followed by a select
  ```
  T1: inline query continuation
  ```
  ```
  (a) from x1 in e1 ... into x2 ...
  -- from x2 in (from x1 in e1 ...) ...
  ```

- Scala: From followed by a join with into followed by a select
  ```
  e1.GroupJoin( e2, 
  x1 => k1, x2 => k2, 
  (x1, g) => e3 )
  ```

Only that the output is Scala syntax (in general, for further processing, as part of *staged rewriting*)

---

Work in progress: *Transforming Scala into a full-fledged DB Programming Language*

Yes, we’re after *Orthogonal Persistence*, in the spirit of PJama or Oberon-D

Only that using a modern, optimizing, database backend (>1 candidate)

Further info: [http://www.sts.tu-harburg.de/people/mi.garcia](http://www.sts.tu-harburg.de/people/mi.garcia)