The Dilemma of Encapsulation vs. Query Optimization

William Tantzen, Jiayi Peng

What is Encapsulation?

- A fundamental of object-oriented programming
  - Hide values of an object to prevent unauthorized access to them.
  - An object should only be able to access other objects that are directly related to it.
  - An indirectly related object should only be accessed by the methods of other intervening objects.

- Examples:
  - Private data members of a class
  - Getter and setter methods
What is Query Optimization?

- DBMSs (Database Management Systems) apply optimizations to queries to greatly increase the efficiency of a request.
- Broad queries give the DBMS more freedom to perform optimizations.
- These optimizations are performed in both relational DBMSs and Object Oriented DBMSs.
The Dilemma

- Encapsulation can reduce the dependencies between objects in a system and decrease that system’s overall complexity.

- Encapsulation makes a system more robust and resilient to change.

- Query optimization is improved by making broad queries.

- Broad queries do not follow the principle of encapsulation.
The Dilemma (cont’d)

- This issue is most prominent in relational DBMSs.
- Object oriented databases also use query optimizers, so they suffer from the same problem (to a lesser degree).
- This dilemma will become more pronounced in both relational and object-oriented databases as query optimizers are improved.
What Should You Do?

Largely dependent on the application you are working on.

● Normal programming applications and object-oriented applications should lean towards *encapsulation*.

● Relational database applications:
  ○ If the programming is complex, then choose *encapsulation*.
  ○ If the programming is easy and query performance is good, then choose *query optimization*.
  ○ If the programming is easy and query performance is poor, then blend *encapsulation* and *query optimization*.
Final Thoughts

● Remember: There is no right answer. This is ultimately a design decision that depends on many different factors.

● This issue is likely to evolve as query optimization improves. Advice that is good today may not be good tomorrow.
References

● **Article:**

● **Images:**
  ○ [http://www.defit.org/2012/02/encapsulation.html](http://www.defit.org/2012/02/encapsulation.html)
  ○ [http://docs.oracle.com/cd/B28359_01/server.111/b28274/optimops.htm#PFGRF001](http://docs.oracle.com/cd/B28359_01/server.111/b28274/optimops.htm#PFGRF001)
  ○ [https://www.osomac.com/2012/12/03/shelfmenu-dilemma/](https://www.osomac.com/2012/12/03/shelfmenu-dilemma/)