

# Re: OOP/OOD Philosophy

---

*Source:* <http://coding.derkeiler.com/Archive/General/comp.object/2005-07/msg00413.html>

---

- *From:* "frebe" <[fredrik\\_bertilsson@xxxxxxxxxxx](mailto:fredrik_bertilsson@xxxxxxxxxxx)>
  - *Date:* 11 Jul 2005 09:09:43 -0700
- 

> Because if you don't Oracle will print something you don't want to see.  
The database should not print anything at all. Your application should print it.

> TO\_DATE and TO\_CHAR are very important in Oracle, yet I've never seen  
> them in any other Standard SQL database.  
You shouldn't use to\_date and to\_char. Retrieve the dates as date objects from the result set.

>> Almost all databases gives you the possibility to create your own  
>> functions.  
> Great but this discussion was about Standard SQL (and lack there of).  
This discussion is about how to make your application switchable between different database vendors. ANSI SQL is obviously important for this, but some problems can be solved in other ways, without having to separate all database access in a separate layer. As a matter of fact, there are an ANSI standard for stored procedures but it is poorly supported.

> Having separate DDL statements (in  
> a script I suppose?) is ugly duplication. The schema and app code get  
> out of phase.  
If you want your application to be able to change the schema, you need a component with different implementations for different vendors. Actually I already did it in Butler.

> "select  
> fieldBlah from tableBlah" holds too much duplication, the field names  
> and table name will be sprayed all over the place ...  
If you have the same SQL statements or partly the same statement, you should of course put it into a separate function/procedure/method. But in a normal enterprise application there are many SQL statements that are only used from one point in the application. In those cases, decoupling will cause you extra work. Saying "the field names and table name will be sprayed all over the place" is the same as saying "the method names and class names will be sprayed all over the place".

If we do it your way, we will sooner or later be forced to use SQL statements that do more things than we want like updating columns that

## Re: OOP/OOD Philosophy

are not changed, joining tables that is not needed and selecting columns we are not interested in.

> In Oracle you use yourSequence.NEXTVAL not NULL to get a new ID. I  
> never thought of trying NULL but I can't see it working.

You are correct. I read the Oracle documentation and they recommended using a trigger. In this case you will still have the same DML (insert) statement for different databases.

> If you just stick closely to "Select \* from Blah" or

> provide a truck load of workarounds yes you should be ok.

When we converted our HR application from Informix to also support Oracle, the didn't have to do this. Just a small number of "workarounds". Our application was mainly written in VB6 and other obsolete languages using a 2-tier architecture. We had some java-code with (switchable) data access object. The 2-tier code did not give us almost any problems at all and we never had to use the different implementations of the DAOs. That DAO-part of the architecture was unnecessary.

> I forgot what MySQL does or doesn't do but I know it's different to

> Oracle and Oracle is different to SQL Server and Hypersonic and ...

If the correct transaction isolation level is chosen and a select statement is made, every record that match to the where-clause should be locked for writing. I think many vendors supports this.

> Transactions do not prohibit race conditions across multiple statements

> only locking can do that.

Transactions and locking work together. You can't have locking without transactions. In the end of the transaction, the locks are released.

Fredrik Bertilsson

<http://butler.sourceforge.net>

.

---

### • *Follow-Ups:*

◆ **Re: OOP/OOD Philosophy**

◇ *From:* Alvin Ryder

### • *References:*

◆ **Re: OOP/OOD Philosophy**

◇ *From:* frebe

◆ **Re: OOP/OOD Philosophy**

◇ *From:* Philip

◆ **Re: OOP/OOD Philosophy**

◇ *From:* Michael Feathers

◆ **Re: OOP/OOD Philosophy**

◇ *From:* frebe

- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Philip
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: frebe
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Philip
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: frebe
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Alvin Ryder
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: frebe
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Alvin Ryder
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: frebe
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Alvin Ryder
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: frebe
- ◆ **Re: OOP/OOD Philosophy**  
    ◇ From: Alvin Ryder

- Prev by Date: **Re: Test first as specification**
- Next by Date: **Re: OOP/OOD Philosophy**
- Previous by thread: **Re: OOP/OOD Philosophy**
- Next by thread: **Re: OOP/OOD Philosophy**
- Index(es):
  - ◆ **Date**
  - ◆ **Thread**