

ActiveRecord Oracle enhanced adapter

Raimonds Simanovskis

Ruby on Rails 2.x => Oracle

```
gem install activerecord-oracle_enhanced-adapter
```

database.yml

development:

```
adapter: oracle_enhanced
```

```
database: XE
```

```
username: blogdemo
```

```
password: blogdemo
```


Oracle Data Types

Ruby	Rails	Oracle
Fixnum	:integer	NUMBER
Float	:float	NUMBER
BigDecimal	:decimal	NUMBER, DECIMAL
Time	:datetime	DATE
Time	:time	DATE
Date	:date	DATE
String	:string	VARCHAR2
String	:text	CLOB
True/FalseClass	:boolean	NUMBER(1), CHAR(1)

Workarounds

```
ActiveRecord::ConnectionAdapters::OracleEnhancedAdapter.
```

```
  emulate_dates_by_column_name = true
```

```
hire_date # Oracle DATE => Ruby Date
```

```
hire_time # Oracle DATE => Ruby Time
```

```
set_date_columns :started_on, :finished_on
```

```
set_datetime_columns :start_date_and_time, :end_date_and_time
```

```
ActiveRecord::ConnectionAdapters::OracleEnhancedAdapter.
```

```
  emulate_booleans_from_strings = true
```

```
active_flag # Oracle CHAR(1) 'Y'/'N' => true/false
```

```
set_boolean_columns :some_boolean_column
```

```
ignore_table_columns :attribute1, :attribute2
```


Calling PL/SQL from Ruby

```
require "oci8"  
conn = OCI8.new("hr", "hr", "xe")  
  
cursor = conn.parse <<-EOS  
BEGIN  
    :return := test_uppercase(:p_string);  
END;  
EOS  
cursor.bind_param(':p_string', "xxx", String)  
cursor.bind_param(':return', nil, String, 4000)  
cursor.exec  
puts cursor[:return]  
cursor.close
```


ruby-plsql gem

```
gem install ruby-plsql
```

```
require "ruby-plsql"  
plsql.connection = OCI8.new("hr", "hr", "xe")  
  
puts plsql.test_uppercase('xxx')
```


ruby-plsql gem

```
plsql.connection = OCI8.new("hr", "hr", "xe")

plsql.test_uppercase('xxx') # => "XXX"
plsql.test_uppercase(:p_string => 'xxx') # => "XXX"
plsql.test_copy("abc", nil, nil) # => { :p_to => "abc",
# :p_to_double => "abcabc" }
plsql.test_copy(:p_from => "abc", :p_to => nil, :p_to_double => nil)
# => { :p_to => "abc",
# :p_to_double => "abcabc" }
plsql.hr.test_uppercase('xxx') # => "XXX"
plsql.test_package.test_uppercase('xxx') # => 'XXX'
plsql.hr.test_package.test_uppercase('xxx') # => 'XXX'

plsql.logoff
```


Custom CRUD methods

```
class Employee < ActiveRecord::Base
  set_create_method do
    plsql.employees_pkg.create_employee(
      :p_first_name => first_name,
      :p_last_name => last_name,
      :p_employee_id => nil
   )[:p_employee_id]
  end
  set_update_method do
    plsql.employees_pkg.update_employee(
      :p_employee_id => id,
      :p_first_name => first_name,
      :p_last_name => last_name
    )
  end
  set_delete_method do
    plsql.employees_pkg.delete_employee(
      :p_employee_id => id
    )
  end
end
```


Other features

- `composite_primary_keys` support
- `NLS_DATE_FORMAT` independent
- `TIMESTAMP` columns support
- Faster schema dump

Thank You!

<http://blog.rayapps.com>

<http://github.com/rsim>