Easy Driver

- Java library with a related code generator, which creates the necessary classes in order to query a relational database.

High Level

- Higher level than JDBC
- No String use and less run-time errors

Small is Beautiful

- It is a small layer
- It is an easy tool to use
- It is not in competition with Hibernate or TopLink

Who unbind you from somebody, binds you to himself.

- There are some databases such as PostgreSQL, which are free and they try to adhere to the standards.

- The generic ORM do not fully use the database functionality.

Indipendent, but happy ?

- Has EJB QL all the functionalities that you need ?
- Do not you miss a function that extracts a month from a date in a query?
- Are you paid in order to study a new sintax?

The boss is in a hurry

- In many contextes the most important thing is to produce something.
- A beautiful architecture takes time, that you could not get.
- The boss is not always right, but you have to understand him.

Targets

- Let us come back to semplicity.
- To change the language should be easy.

Will the community prefere Java in the future or will it become a choice for the big corporate only? How much does it cost to migrate to Python or C++ ?

Idea 1: a class represent a table metadata.

Idea 2: the hard and boring work is made by a code generator.

Where do we go ?

Compatibility with SQLite and porting to C++ and Objective C

Easy Driver is looking for help

- Are we helping a possibile competitor, creating a code generator ?

- Maybe, but even competitors can help each other, sometimes, in order to get rid of some uncomfortable suppliers...

Is Easy Driver credible ?

- Of course it is not complete, but there is a working base, to cooperate could mean to show the problems and to suggest some new developments.
- The credibility of a project is given by people who use it. As more you help me, more the project will be credible.

My tool is enough for me!

- There are no Swiss knifes, there is place for another tool in the niches which are left by the main frameworks.
- In the industrial boards, in the embedded hardware or mobile, a ligth tool helps...

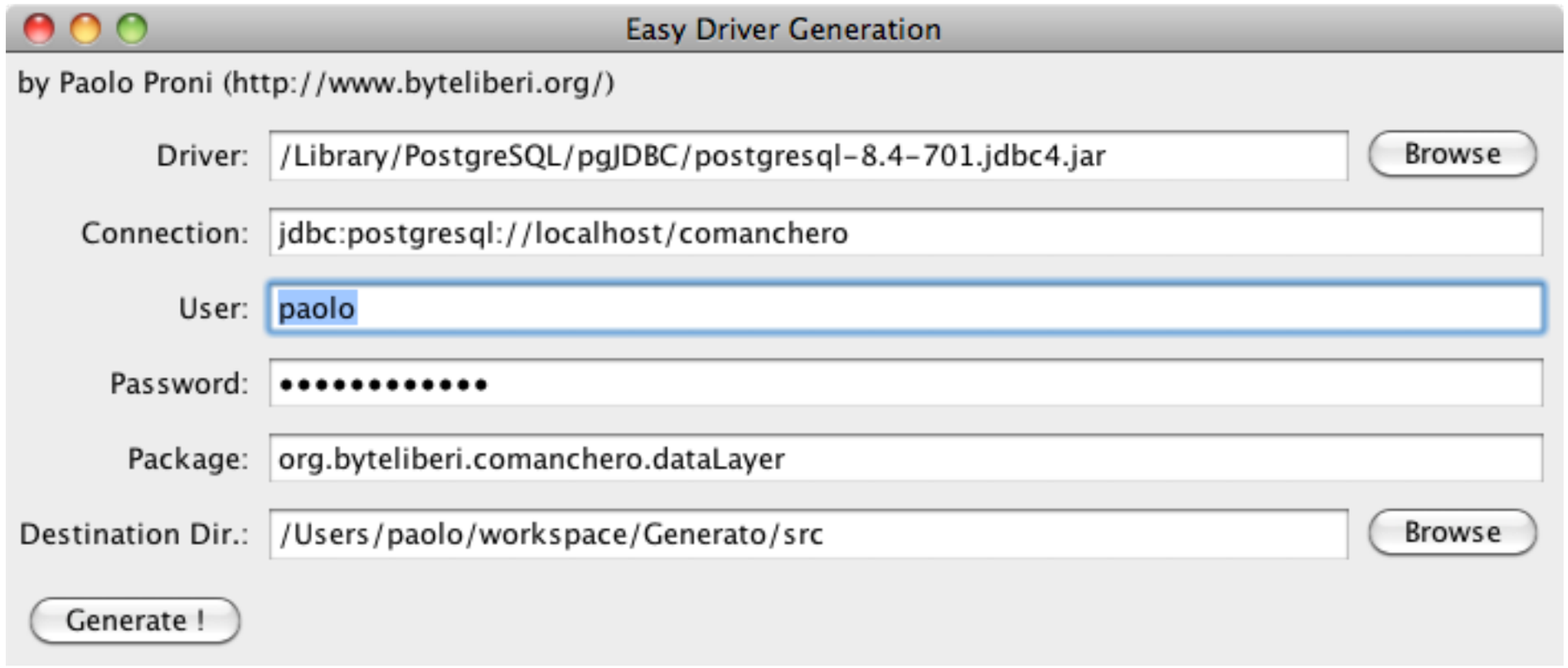
Will the customers like it ?

- What advantage the actual customer can have ?
- It is a good way to make demo in an agile way, anyway it could help to find new customers.

General Public License

- Could LGPL be a better choice?
- The Linux kernel diffusion has not been slowed by its license. I invite who do a derivative work to apply the same license, else they should contact me, the goal is to know more people...

The generator



The structure

```
package org.byteliberi;

import org.byteliberi.easydriver.*;
import org.byteliberi.easydriver.fields.*;

public enum TabellaB {
    INSTANCE;

    private DBTable table;
    private IntField id;
    private VarcharField vc;

    private TabellaB() {
        this.table = new DBTable("tabella_b");
        this.id = new IntField("id", false, table);
        this.vc = new VarcharField("vc", true, table);
        this.table.setPrimaryKey(new PrimaryKey(this.id));
    }

    public final DBTable getTable() {
        return table;
    }

    public final IntField getId() {
        return id;
    }

    public final VarcharField getVc() {
        return vc;
    }

}
```

Object Model

```java
package org.byteliberi;

public class TabellaBObjectModel {

    private Integer id;
    private String vc;

    public TabellaBObjectModel() {
    }

    public TabellaBObjectModel(final Integer id) {
        this.id = id;
    }

    public final Integer getId() {
        return id;
    }

    public final String getVc() {
        return vc;
    }

    public final void setId(final Integer id) {
        this.id = id;
    }

    public final void setVc(final String vc) {
        this.vc = vc;
    }
}
```

Factory

```java
package org.byteliberi;

import java.sql.ResultSet;
import java.sql.SQLException;
import org.byteliberi.easydriver.ObjectFactory;

public class TabellaBObjectModelFactory implements ObjectFactory<TabellaBObjectModel> {

    public TabellaBObjectModelFactory() {
    }

    @Override
    public final TabellaBObjectModel map(final ResultSet rs) throws SQLException {
        final TabellaB table = TabellaB.INSTANCE;
        final TabellaBObjectModel vo = new TabellaBObjectModel();
        vo.setId( table.getId().map(rs, 1) );
        vo.setVc( table.getVc().map(rs, 2) );
        return vo;
    }
}
```

Service Structure

```
public final TabellaBObjectModel selectByPK(final Connection con, final Integer id) throws SQLException {
        final TabellaB tableStruct = TabellaB.INSTANCE;
        final SelectQuery<TabellaBObjectModel> query =
                                new SelectQuery<TabellaBObjectModel>(TabellaB.INSTANCE.getTable(),
                                new TabellaBObjectModelFactory());
        query.setWhere(new Equals(tableStruct.getId()));
        query.prepareQuery(con);
        query.addParameter(id);
        return query.getSingleResultAndClose();
    }

    public final int deleteByPK(final Connection con, final Integer id) throws SQLException {
        final TabellaB tableStruct = TabellaB.INSTANCE;
        final DeleteQuery query = tableStruct.getTable().createDeleteQuery();
        query.setWhere(new Equals(tableStruct.getId()));
        query.prepareQuery(con);
        query.addParameter(id);
        return query.execute();
    }

    public final int insert(final Connection con, final TabellaBObjectModel model) throws SQLException {
        final TabellaB tableStruct = TabellaB.INSTANCE;
        final InsertQuery query = tableStruct.getTable().createInsertQuery();
        query.prepareQuery(con);
        query.addParameter(model.getId());
        query.addParameter(model.getVc());
        return query.execute();
    }

    public final int updateByPK(final Connection con, final Integer id, final TabellaBObjectModel model)
                                throws SQLException {
        final TabellaB tableStruct = TabellaB.INSTANCE;
        final UpdateQuery query = tableStruct.getTable().createUpdateQuery();
        query.setWhere(new Equals(tableStruct.getId()));
        query.prepareQuery(con);
        query.addParameter(model.getId());
        query.addParameter(model.getVc());
        query.addParameter(id);
        return query.execute();
    }
```

http://www.byteliberi.org/