Easy Driver

- Libreria Java con un generatore di codice ad essa associato, che crea le classi necessarie per accedere ad un database relazionale.

High Level

- Livello più alto di JDBC
- Niente uso di stringhe e meno errori a run-time.

Piccolo è Bello

- E' uno strato piccolo
- E' uno strumento semplice da usare
- Non è in competizione con Hibernate o TopLink

Chi ti slega da un altro, lo fa per legarti a sé stesso.

- Ci sono dei database come PostgreSQL, che sono liberi e cercano di aderire il più possibile agli standard.

- Gli ORM generici non sfruttano pienamente la funzionalità di un database.

Indipendenti, ma contenti?

- EJB QL ha le funzionalità che vi servono ?
- Non sentite la mancanza di una funzione che estragga il mese da una data dentro ad una query?
- Vi pagano lo studio di nuove sintassi?

Il capo ha fretta

- In molti contesti la cosa più importante e produrre presto qualcosa.

- Una bella architettura richiede tempo, che si può chiedere, mai pretendere.

- Il capo non ha sempre ragione, però va capito.

Obbiettivi

- Ritorno alla semplicità.
- Facilità nel cambio di linguaggio di programmazione.

Java sarà il linguaggio preferito dalla community anche in futuro o si orienterà verso le grandi corporate? Quanto costa migrare in Python o C++ ?

Idea 1: una classe rappresenta i metadati di una tabella.

Idea 2: il lavoro duro e noioso lo fa un generatore.

Dove si va ?

Compatibilità con SQLite e porting verso C++ e Objective C

Easy Driver cerca aiuto

- Aiutare un generatore di codice può aiutare un consulente oppure un'azienda concorrente ?

- Può darsi, però anche i concorrenti possono aiutarsi a volte, per slegarsi da fornitori scomodi...

Easy Driver è credibile ?

- Sicuramente non è completo, ma c'è una base funzionante, collaborare può voler dire anche solo segnalare i difetti e proporre nuovi sviluppi.
- La credibilità di un progetto è data dalla gente che partecipa, più aiutate, più sarà credibile.

A me basta il mio strumento!

- Non esistono i coltellini svizzeri, c'è posto per un altro strumento nelle nicchie lasciate aperte dai framework principali.
- Nelle schede industriali, nell'hardware embedded o mobile, uno strumento leggero, aiuta...

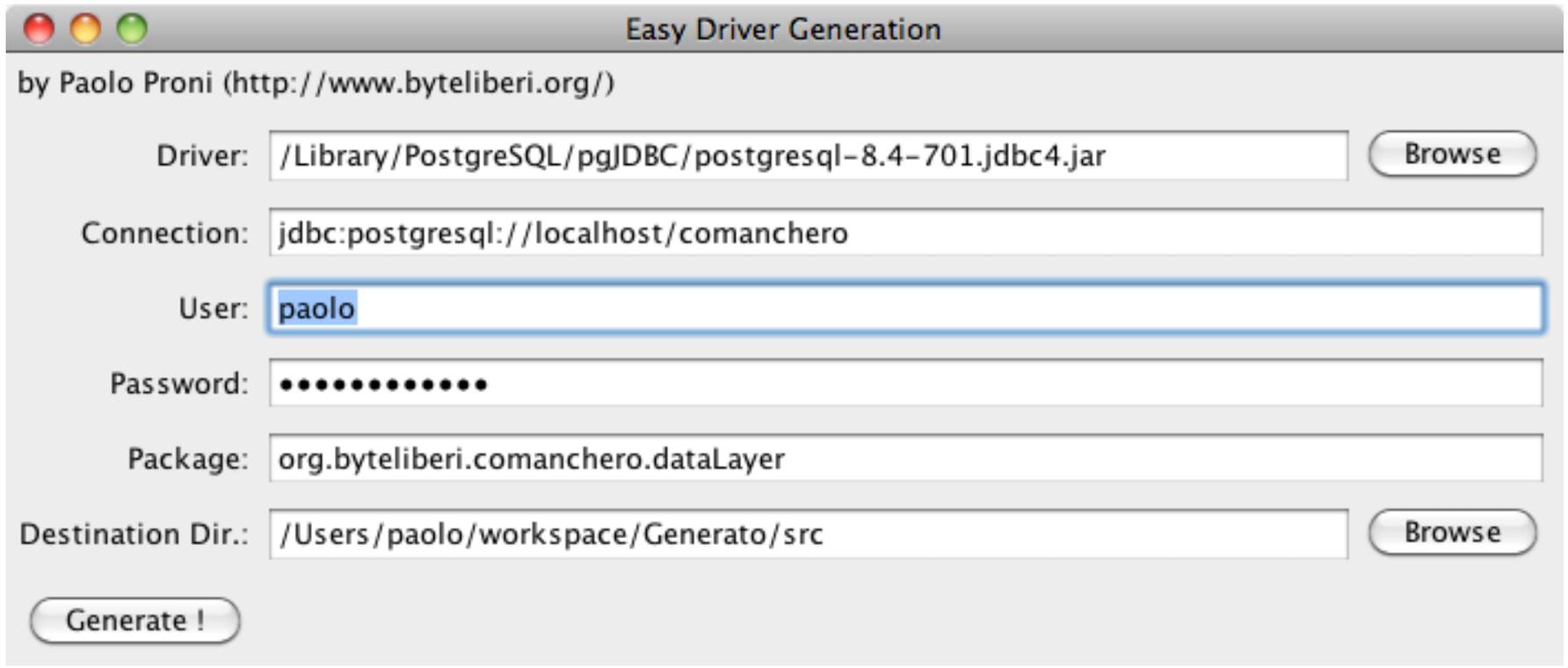
Ai clienti piacerà ?

- Che vantaggio avrebbero i clienti attuali ?
- E’ un buon modo per fare demo in modo agile, comunque fare presto aiuta a trovarne dei nuovi.

General Public License

- Sarebbe meglio LGPL?
- La diffusione del kernel di Linux non è stata ostacolata dalla licenza. Si invita chi fa un lavoro derivato ad applicare la stessa licenza, altrimenti mi si contatti almeno, l'obbiettivo è fare conoscenza...

Il generatore



La struttura

```java
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;
    }
}
```

Struttura a Servizi

```
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/