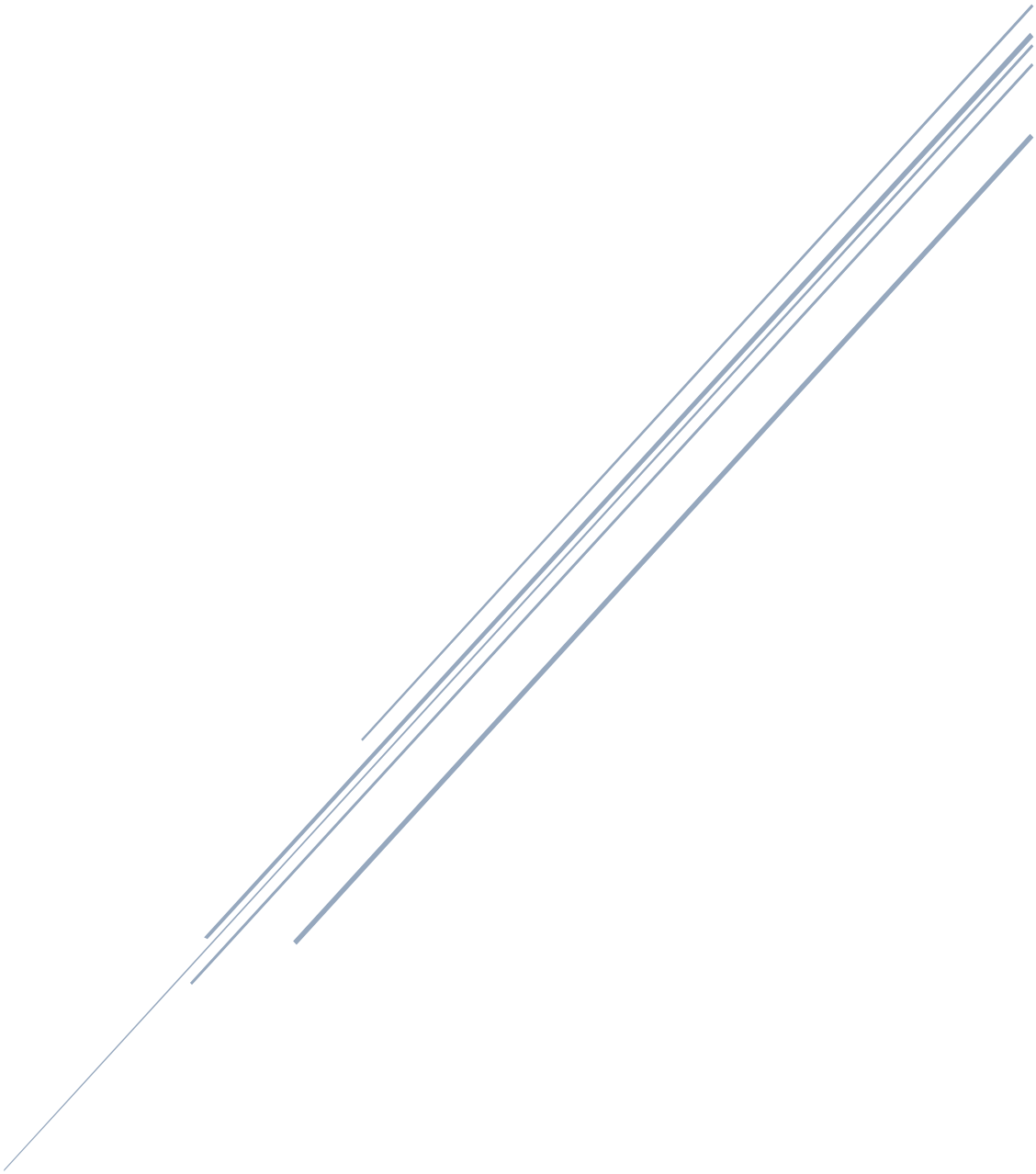
PROGETTO DI BASI DI DATI 2

Centro commerciale

Guida al deploy



Francesco Pontillo
Matricola 600119

Sommario

1 Creazione dei database

1.1 Prerequisiti

Le basi di dati sono state realizzate utilizzando PostgreSQL 9.2.

La configurazione utilizzata sulla macchina dove il progetto è stato realizzato è la seguente:

- Hostname: `localhost`
- Porta: `5432`
- User: `postgres`
- Password: `30lode`

1.2 Database per l'applicativo Web: db2

1. Creare un nuovo database con nome `db2` e assegnare l'utente `postgres` come proprietario.
2. Connettersi al database appena creato ed eseguire lo script SQL `db/scripts.sql`.
3. Il database è stato creato con alcuni dati di prova.

1.3 Database per l'analisi: db2-dw

1. Creare un nuovo database con nome `db2-dw` e assegnare l'utente `postgres` come proprietario.
2. Connettersi al database appena creato ed eseguire lo script SQL `dw/scripts.sql`.
3. Il database è stato creato con alcuni dati di prova.

2 Configurazione del servlet container

2.1 Prerequisiti

Il Web server utilizzato è Tomcat 7.

La configurazione utilizzata sulla macchina dove il progetto è stato realizzato è la seguente:

- Hostname: `localhost`
- Porta: `8080`
- Username: `admin`
- Password: `30lode`
- Percorso di deploy: `${catalina.home}/webapps`

2.2 JDBC Connector

Per utilizzare Mondrian tramite jPivot è necessario esporre a tutte le applicazioni (quindi anche a jPivot) la libreria per eseguire connessioni di tipo JDBC ai database PostgreSQL: copiare il file `tomcat-conf/postgresql-9.2-1002.jdbc4.jar` nella cartella delle librerie comuni di Tomcat.

Nella configurazione standard di Tomcat 7, una delle directory gestite in maniera condivisa è `${catalina.home}/lib`.

2.3 Esposizione dei DataSource

Per poter fornire alle applicazioni l'accesso ai database si impostano due fonti di dati (DataSource) nel file `${catalina.home}/conf/context.xml`. Le fonti di dati di nostro interesse sono incluse nel fil `tomcat-conf/context.xml`: ricopiarle nel file di configurazione di Tomcat sotto il tag `Context`.

3 Compilazione del servizio RESTful

La compilazione del servizio RESTful è necessaria solo se il nome delle sorgenti dati (DataSource) impostate al punto 2.3 variano. Se i loro nomi non sono stati modificati, si può evitare di modificare e ricompilare il servizio Web. In tal caso, si può passare al punto 4.

3.1 Prerequisiti

È stato utilizzato Eclipse Juno per la creazione del servizio Web.

3.2 Generazione delle classi di jOOQ

La generazione delle classi è opzionale, e serve solo se è stato modificato lo schema di `db2`. Se vengono rigenerate le classi per un diverso schema servirà fare opportuni cambiamenti al servizio RESTful (progetto in `rest`), che si basa su tali classi.

1. Importare in Eclipse il progetto in `rest-gen`.
2. Se non sono stati utilizzati i parametri di default per il database, modificare opportunamente il file `src/jooq.xml`.
3. Avviare il progetto in Eclipse, le nuove classi verranno generate e aggiunte al progetto in `rest`.

3.3 Compilazione del servizio RESTful

1. Importare in Eclipse il progetto in `rest`.
2. Se le classi jOOQ sono state rigenerate, correggere eventuali problemi ed apportare gli opportuni cambiamenti.
3. Il progetto punta automaticamente alla sorgente dati `jdbc/MallDB`, definita al punto 2.3. Se il nome di tale sorgente è stato modificato, modificare opportunamente il riferimento nel file `rest/WebContent/WEB-INF/web.xml`.
4. Compilare il progetto.
5. Il progetto è pronto per l'esportazione.
    a. Selezionare il progetto.
    b. File → Export → WAR File.
    c. Selezionare come percorso del file `rest/war/rest.war`.

4 Deploy del servizio RESTful

1. Se Tomcat non è ancora avviato, avviarlo.
2. Andare su http://localhost:8080/manager/html ed inserire le credenziali di login.
3. Eseguire il deploy del file `rest/war/rest.war`.
4. Il servizio Web è caricato e raggiungibile alla URL http://localhost:8080/rest/.

5 Configurazione di jPivot

5.1 Prerequisiti

È stato utilizzato jPivot alla versione 1.8.0. jPivot deve essere caricato come `.war` all'interno di Tomcat 7. jPivot deve essere già caricato come `war` in Tomcat.

5.2 Aggiunta della sorgente dati

jPivot deve essere in grado di connettersi alla sorgente dati che andremo a specificare nella query.

1. Aprire il file `web.xml` di jPivot (`${catalina.home}/webapps/jpivot/WEB-INF/web.xml`).
2. Aggiungere nel tag `web-app` la risorsa definita in `dw/web.xml`.

5.3 Aggiunta delle pagine di visualizzazione

Per poter interagire con gli oggetti messi a disposizione di jPivot, si devono copiare lo schema del cubo, la pagina di query e una pagina di template nella cartella di jPivot su Tomcat:

1. Copiare i file `dw/mallSchema.xml` e `dw/mallQuery.jsp` in `${catalina.home}/webapps/jpivot/WEB-INF/queries`.
2. Copiare il file `dw/better.jsp` in `${catalina.home}/webapps/jpivot/`.

6 Configurazione e building dell'applicazione Web

Se nessuno dei seguenti indirizzi è stato modificato dai default suggeriti in questa guida, si può procedere al punto successivo:

- Indirizzo dei servizi RESTful, default: http://localhost:8080/rest
- Indirizzo della pagina di jPivot: http://localhost:8080/jpivot/better.jsp?query=mallQuery

Altrimenti è necessario seguire i seguenti passi per configurare e creare una build dell'applicazione Web.

6.1 Installazione di Yeoman

Yeoman è un tool per la gestione di progetti applicativi per il Web. Mentre al momento d'inizio dello sviluppo era presente solo la versione 0.9.6, adesso è diventata disponibile la versione 1.0 beta, che non è compatibile con il progetto corrente.

Scaricare yeoman alla versione 0.9.6 (istruzioni sul sito http://yeoman.io/installation.html, è stato testato su Mac OS X).

6.2 Configurazione delle connessioni

Aprire il file `web/app/scripts/services/Config.js` e modificare le variabili:

- `baseAPI`, impostare la URL (i due punti ":" sono preceduti dalla stringa di escape "\\") del servizio Web, seguita dalla stringa "/api/". Non modificare se il percorso non è cambiato.
- `analysisPage`, impostare la URL completa per la visualizzazione della pagina di analisi tramite jPivot. Qui non serve eseguire l'escaping dei due punti.

6.3 Test senza building

Passo opzionale.

Una volta modificato il file di configurazione si può provare il progetto senza eseguire il building. Spostarsi da riga di comando nella cartella `web` e lanciare il comando `_server.sh` (o `_server.bat` su sistemi Windows). A meno di errori nella configurazione dei path di yeoman o particolari differenze nei sistemi, dovrebbe avviarsi un piccolo server Web che apre una pagina nel browser predefinito in cui è possibile testare l'applicativo.

In caso di errori, sempre nella cartella `web`, digitare il comando `yeoman server`, che produrrà lo stesso risultato.

Terminato il testing, premere CTRL-C nello script lanciato per chiudere il processo server.

6.4 Building

Spostarsi da riga di comando nella cartella `web` e lanciare il comando `_build_war.sh` (o `_build_war.bat` su sistemi Windows), che scompone il processo di building in due fasi:

1. Building da parte di Yeoman.

2. Compattazione in un archivio `mall.war`, che verrà posizionato in `web/war`.

Nel caso in cui il comando dovesse andare in errore, procedere al building ed alla compattazione manuale tramite i comandi, da eseguire sempre nella cartella `web` e in ordine:

1. `yeoman build`
2. `jar cvf ../war/mall.war *`

Al termine dell'esecuzione sarà disponibile l'archivio `web/war/mall.war`.

7 Deploy dell'applicazione Web

1. Se Tomcat non è ancora avviato, avviarlo.
2. Andare su http://localhost:8080/manager/html ed inserire le credenziali di login.
3. Eseguire il deploy del file `web/war/mall.war`.
4. L'applicazione Web è caricata e raggiungibile alla URL http://localhost:8080/mall/.

8 Utilizzo e note

L'applicazione ha un unico utente amministratore, con le seguenti credenziali:

- Username: `admin`
- Password: `30lode`

La password è memorizzata in `db2` tramite cifratura SHA-256.

L'applicazione Web è stata testata su Chrome e Safari.

È stato eseguito il deploy dei pacchetti anche sulla piattaforma cloud OpenShift. L'applicativo Web è raggiungibile all'indirizzo https://bd2-frapontillo.rhcloud.com/mall, a cui si può accedere con le stesse credenziali. Per il deploy su OpenShift è stato necessario modificare gli script di creazione degli schemi anticipandone l'esecuzione con quella del comando: `CREATE LANGUAGE plpgsql`. Ciò è dovuto al fatto che OpenShift supporta PostgreSQL alla versione 8.4. In questo modo è stato possibile eseguire funzioni scritte anche in PL/pgSQL.