# INFORME FINAL DE PRÀCTIQUES

Realitzar una comparativa de rendiment i utilització entre els diferents models de bases de dades orientades a columnes mitjançant la construcció i explotació d'un cub OLAP utilitzant la suite de BI Pentaho.

El Pentaho BI server es una eina open source que te un conjunt d'eines de Business intel·ligències que permeten des de la carrega de dades i transformació d'aquestes, passant per l'anàlisi mitjançant cubs OLAP i com a resultat donant una serie de respostes al usuari en forma de taules pivotants, informes dinàmics i gràfics dins d'una consola d'usuari, en un entorn integrat i totalment reconfigurable.

Aquest servidor estarà connectat a un servidor Mysql que es un sistema gestor de bases de dades relacionals allotjat a la mateixa màquina que albergarà les dues bases de dades que utilitzarem per la comparativa.

Els objectius per a dur a terme el projecte amb èxit son els següents:

- Instal·lar i configurar un servidor pentaho i Mysql en una màquina.
- Crear les dues bases de dades a comparar.
- Dissenyar els respectius cubs de dimensions
- Extreure la informació necessària per extreure les conclusions

Joan Piferrer Huerta

---

Primer de tot necessitarem instal·lar el pentaho bi server. Aquest servidor aixecara un servidor tomcat amb una infraestructura de tipus portal que albergarà el motor de procés dels cubs multi-dimensionals (mondrian) i donarà suport als usuaris que hi tinguin accés per poder analitzar i publicar les dades amb una sèrie d'eines que proporciona el portal.

A priori, la instal·lació es d'allò més senzilla, només hem de tenir un servidor amb el sdk de java i fer que l'executable d'arrancada del servidor tingui el java_path apuntant a el directori corresponent.

Un cop fet això tindrem el servidor funcionant, però per defecte el servidor apunta a un HSQLDB. Aquest sistema no ens serveix ja que no es un sistema persistent en memòria sino que crea les bases de dades cada cop que la màquina arranca.

Per fer que el servidor s'alimenti de un sistema gestor diferent hem de canviar una sèrie de paràmetres perquè el servidor apunti a un MySql allotjat a la mateixa màquina amb unes credencials determinades. El procés que he seguit per dur a terme aquesta configuració està enllaçat a la bibliografía.

Un cop fet això s'ha de crear una base de dades que s'anomena hibernate per la configuració i una que s'anomena quarz encarregada d'emmagatzemar les tasques programades.

També cal afegir els drivers de Mysql a les carpetes de llibreries corresponents.

Un cop fet tot això executarem l'arxiu d'arrancada dels serveis del servidor i de la consola d'administració del pentaho biserver i testejarem els informes de proves per comprovar que tot funciona correctament.

Una fase molt important del projecte es la selecció i creació dels dos models de bases de dades que utilitzarem. Com ja hem comentat abans seran un model d'única taula i un model en estrella.

Mysql té a disposició dels usuaris una sèrie de bases de dades de mostra que poden ser descarregades i utilitzades per fer tot tipus de proves. Necessitem una base de dades ni molt gran ni molt petita i de les que hi ha penjades a la web hem seleccionat la anomenada employees.

Per dur a terme la creació i bolcat de les dades de la base de dades employees utilitzarem la eina mysql workbench d'oracle.

L'altre base de dades amb un tipus de model d'única taula la crearem a partir de la d'estrella a partir d'una consulta que engloba totes les taules fent left outer joins. El resultat d'aquesta consulta la inserirem en una taula molt gran que inclourà totes les dades de l'anterior model.

Un cop tenim la taula d'aquesta manera haurem de fer dues vistes per extreure les taules necessaries que ens serviran per dissenyar la dimensio departament i per calcular la suma d'empleats

Un cop arribats a aquest punt haurem de dissenyar els cubs que utilitzarem alhora d'analitzar les dades i fer les proves de rendiment. Dissenyarem un cub que ens permeti comptar empleats separats per departaments.

Per comprendre perque s'anomena cub olap o cub de dimensions veiem en la següent figura un exemple de cub on tenim la dimensio temps a la dreta, la dimensio categories a l'esquerra i les mesures a baix. Per tant podem saber el marge de benefici per exemple, agrupat per temps i per categoria de productes, perque el cub esta dissenyat amb les

mesures i dimensions que ens permeten fer-ho.

Així tenim que haurem de crear una dimensió departaments i una mesura empleats i així poder utilitzar-les en el Jpivot més tard. Un cop fet el publicarem al servidor pentaho sota la contrasenya de publicació definida en els paràmetres de configuració del pentaho biserver.

Aquesta operació la repetirem per cadascun dels dos models que tenim i els publicarem sota noms diferents.

El disseny el durem a terme amb la eina Pentaho schema workbench que és un entorn de desenvolupament que et permet crear i provar cubs OLAP cub de manera visual. El motor Mondrian processa les sol·licituds d'MDX amb els esquemes ROLAP (OLAP Relacional). Aquests arxius d'esquemes són models de metadades XML que es creen en una estructura específica que utilitza el motor de Mondrian. Aquests models XML poden ser interpretats com a estructures en forma de cub que utilitzen taules de dimensions. No es requereix de la persistència en memòria d'un cub físic real.

Arribat a aquest punt només hem de crear un anàlisis de dades dels cubs publicats en el punt anterior per cada model de base de dades a testejar: en estrella i en taula.

Al fer clic al boto “New Analysis” i seleccionar els cubs que necessitem en aquell moment.

Un cop seleccionat el sistema immediatament ens presenta les dades i ja podem començar a jugar amb les dimensions i les mesures:

Podem redefinir en qüalsevol moment les dimensions i mesures que volem utilitzar.

El Jpivot dinàmicament va generant les consultes MDX que corresponen amb les peticions que donem a través de la seva interfície:

Abans de poder mesurar els temps de resposta haurem de modificar els paràmetres de configuració del pentaho per que afegeixi informació addicional als logs que es generen per defecte.

Per poder mesurar el temps de resposta haurem de buscar els logs del servidor pentaho, allà trobarem totes les peticions que ha rebut el servidor i el seu temps de resposta amb mili-segons.

**Model estrella**

```
select
   count(`employees`.`emp_no`) as `m0`
from
   `employees` as `employees`]
```
exec 145 ms

```
select
   `dept_emp`.`dept_no` as `c0`
from
   `dept_emp` as `dept_emp`
group by
   `dept_emp`.`dept_no`
order by
   ISNULL(`dept_emp`.`dept_no`) ASC,
`dept_emp`.`dept_no` ASC]
```
exec 0 ms

```
executing sql [
select
   count(distinct `dept_emp`.`dept_no`) as `c0`
from
   `dept_emp` as `dept_emp`]
```
exec 150 ms

**Model taula**

```
select
   count(`employees`.`emp_no`) as `m0`
from
   `employees` as `employees`]
```
exec 11576 ms

```
select
   `dept_emp`.`dept_no` as `c0`
from
   `dept_emp` as `dept_emp`
group by
   `dept_emp`.`dept_no`
order by
   ISNULL(`dept_emp`.`dept_no`) ASC,
`dept_emp`.`dept_no` ASC]
```
exec 6384 ms

```
executing sql [
select
   count(distinct `dept_emp`.`dept_no`) as `c0`
from
   `dept_emp` as `dept_emp`]
```
exec 6238 ms

Podem distingir perfectament les dues consultes iguals amb temps de resposta molt dispars.

El resultat es veu molt fàcilment, el esquema en estrella funciona molt mes ràpid amb un temps de resposta menor que el esquema en format de taula.