

# Projecte fi de Màster

Comparativa de rendiment OLAP per a diferents  
motors de bases de dades orientades a columnes

Especialitat: Desenvolupament d'aplicacions

Autor: Joan Piferrer Huerta

Consultor: Gregorio Robles Martínez

Consultor extern: Juan José Ortilles

Barcelona, Juny 2013



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

## COPYRIGHT

### Autor

Joan Piferrer Huerta.

Per a qualsevol comentari o suggeriment contactar amb [joanpiferrer@gmail.com](mailto:joanpiferrer@gmail.com)

### Publicació

Contingut publicat el 9 de juny de 2013.

Document creat en LibreOffice 3.6.2.2

### Llicència



La memoria del projecte "Comparativa de rendiment OLAP per a diferents motors de bases de dades orientades a columnes" està sota una **licència de Creative Commons**

**[Reconocimiento-CompartirIgual 3.0 Espanya.](https://creativecommons.org/licenses/by-sa/3.0/es/)**

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

# Comparativa de rendiment OLAP per a diferents motors de bases de dades

## Índex

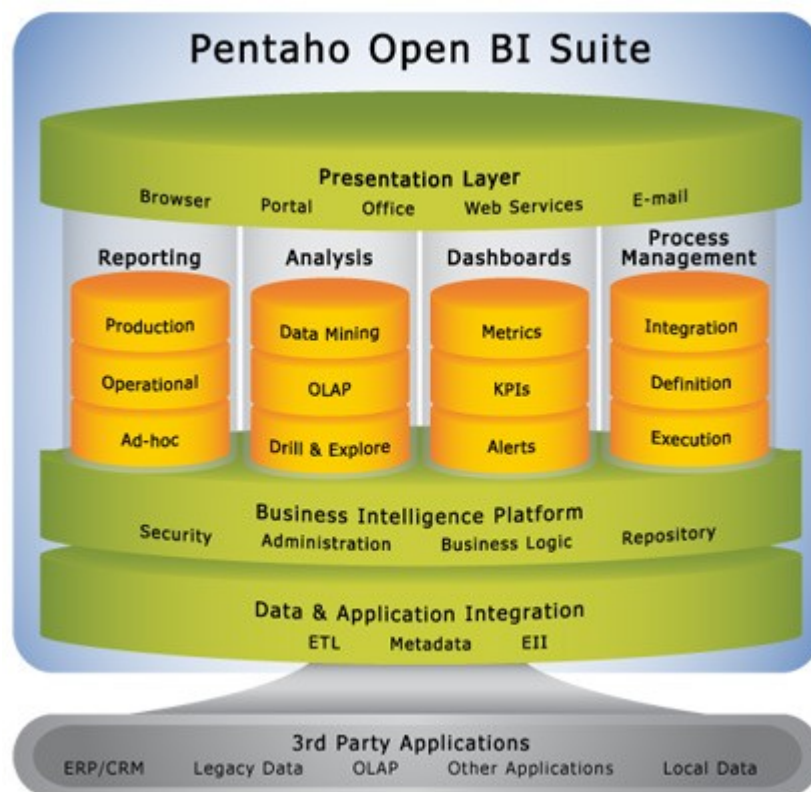
COPYRIGHT.....	3
INTRODUCCIÓ.....	6
REQUISITS.....	8
Eines suite pentaho:.....	8
FITES TEMPORALS.....	9
VIABILITAT.....	10
Viabilitat econòmica.....	10
Viabilitat tècnica.....	11
Viabilitat legal.....	11
DISSENY I IMPLEMENTACIÓ.....	12
Servidor Pentaho.....	12
Bases de Dades.....	14
Cubs de Dimensions.....	19
Anàlisi de Dades.....	21
Monitorització del temps de resposta.....	24
RESULTATS.....	26
CONCLUSIONS.....	31
Objectius aconseguits.....	31
Experiència Personal.....	34
Aprenentatge.....	35
Futures aplicacions i vies de continuació.....	39
BIBLIOGRAFIA.....	43

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

# INTRODUCCIÓ

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.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

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.



# Comparativa de rendiment OLAP per a diferents motors de bases de dades

## REQUISITS

Els requisits identificats 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

Per dur a terme aquests requisits necessitarem, a part del MySQL i el servidor pentaho l'ajuda de les eines de la suite pentaho bi server i del mysql workbench.

### ***Eines suite pentaho:***

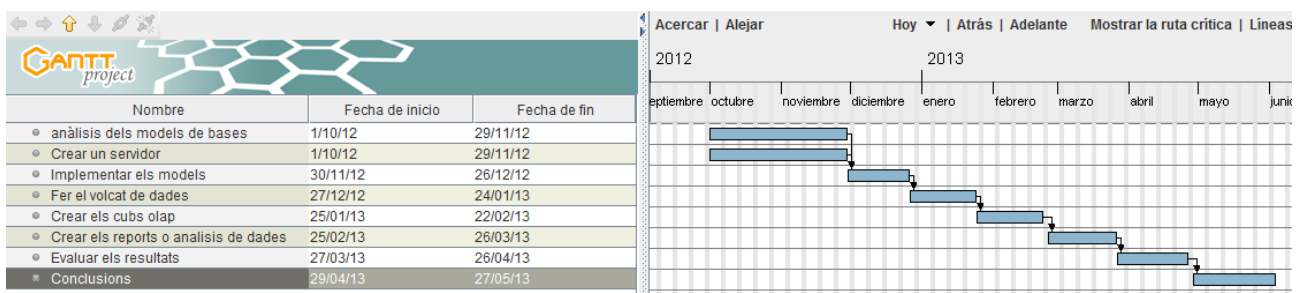
- ETL: eina de carrega i transformació de dades també anomenat Spoon. En aquest projecte no caldrà utilitzar-lo ja que tenim la base de dades en model estrella i en format sql llest per bolcar-lo al Mysql i amb l'ajuda de consultes SQL ens serà suficient per transformar-la a un format de taula única.
- Schema workbench: Programa de creació i gestió de cubs multi-dimensionals al repositori de pentaho.
- Jpivot: Llibreria en jsp que renderitza taules i gràfics de dades allotjades en cubs OLAP.
- Saiku: Eina semblant a l'anterior però més moderna i probable successora d'aquesta.
- CDE: Interfície que renderitza dashboards i gràfics dinàmics.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

### FITES TEMPORALS

- Fer un anàlisi dels models de bases de dades que s'utilitzaran en la comparativa.
- Crear un servidor i instal·lar un mysql i un pentaho biserver
- Implementar els models que s'han de testejar en una base de dades de mysql
- Fer el bolcat de dades en cadascun dels models.
- Crear els cubs olap que es connectaran als diferents models
- Crear els reports o anàlisis de dades on es presentaran les dades filtrades pels cubs.
- Avaluar els resultats obtinguts amb cadascun dels models
- Conclusions



## VIABILITAT

A continuació es mostra un petit estudi de viabilitat del projecte, analitzant aspectes econòmics, tècnics i legals.

### ***Viabilitat econòmica***

Per dur a terme el projecte, no hi ha un pressupost establert però si una durada màxima de 8 mesos. Donada la situació econòmica, es requereix el mínim de recursos per finalitzar amb èxit el projecte. Els recursos inicialment proposats son:

- **Recursos humans**

Comptarem amb la dedicació durant 8 mesos d'un analista programador, que s'encarregarà d'implementar el projecte. També hi haurà la intervenció d'un cap de projecte, que s'encarregarà de supervisar-lo.

- **Eines de treball**

Només s'usarà programari lliure, per tant el cost en llicències serà nul. Es necessita un pc de gamma mitja que accedira al servidor i aquest que estarà implementat en un debian virtualitzat que farà de servidor pentaho amb una capacitat de 8gb de disc dur i 1gb de ram.

- **Altres**

No es contemples altres despeses especials.

**El projecte es perfectament viable econòmicament parlant.**

### ***Viabilitat tècnica***

El principal problema que podem trobar en l'apartat tècnic, es la desconexió d'alguns programari a utilitzar. Això pot retardar la realització d'alguna de les tasques programades. A pesar de tot, com que comptem amb una ampla experiència en altres tecnologies J2EE o MySQL, poden compensar les petites desviacions de temps.

**Podem concloure que el projecte és viable tècnicament.**

### ***Viabilitat legal***

Tenint en compte que el projecte esta completament desenvolupat en programari lliure, només caldrà tenir en compte les limitacions que poden tenir certes clàusules entre els diferents programaris que hi haurà en l'entrega final.

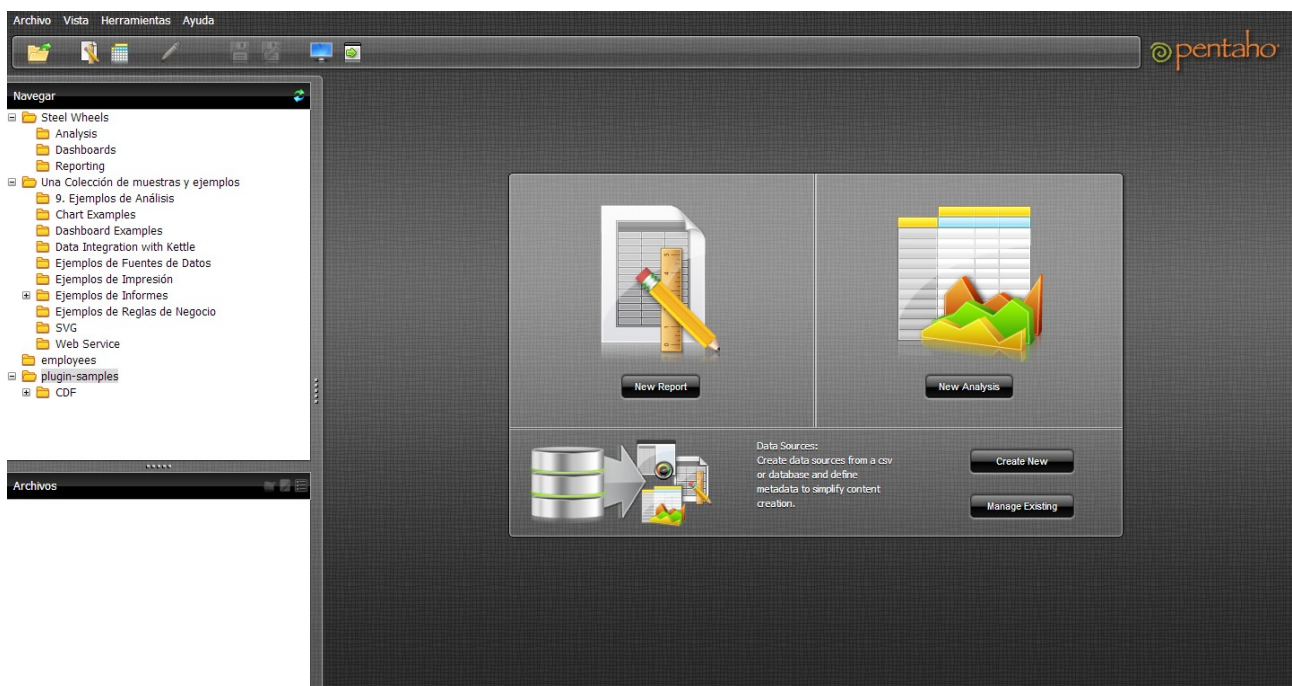
Com que aquest tema te una complexitat elevada, es dedicarà un cert temps del projecte a estudiar possibles incompatibilitats de cara a l'entrega final, eliminant tot aquell software incompatible.

**Així doncs, el projecte és viable legalment.**

## DISSENY I IMPLEMENTACIÓ

### *Servidor Pentaho*

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

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

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 bibliografia.

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

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

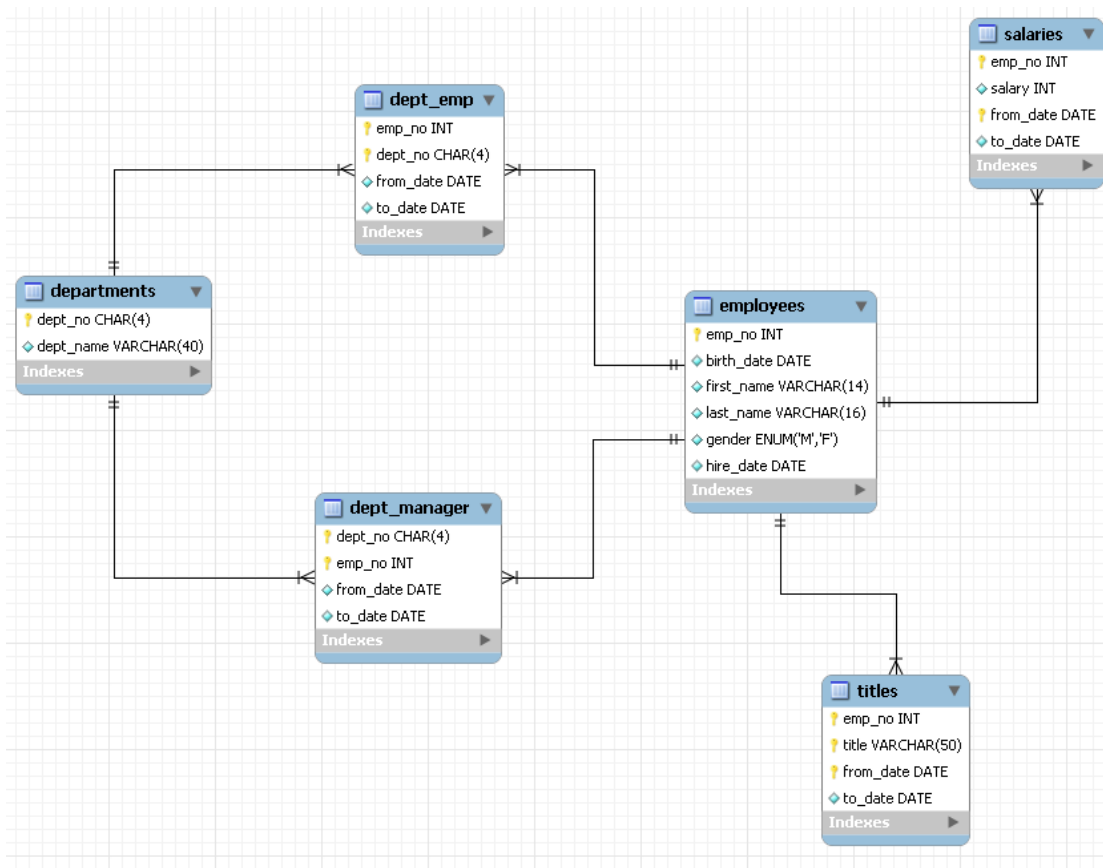
Un cop fet tot això executarem l'arxiu d'arrancada dels serveis del servidor i de la consola d'administració del pentaho biserver i testejarem els informes de proves per comprovar que tot funciona correctament.

# Comparativa de rendiment OLAP per a diferents motors de bases de dades

## Bases de Dades

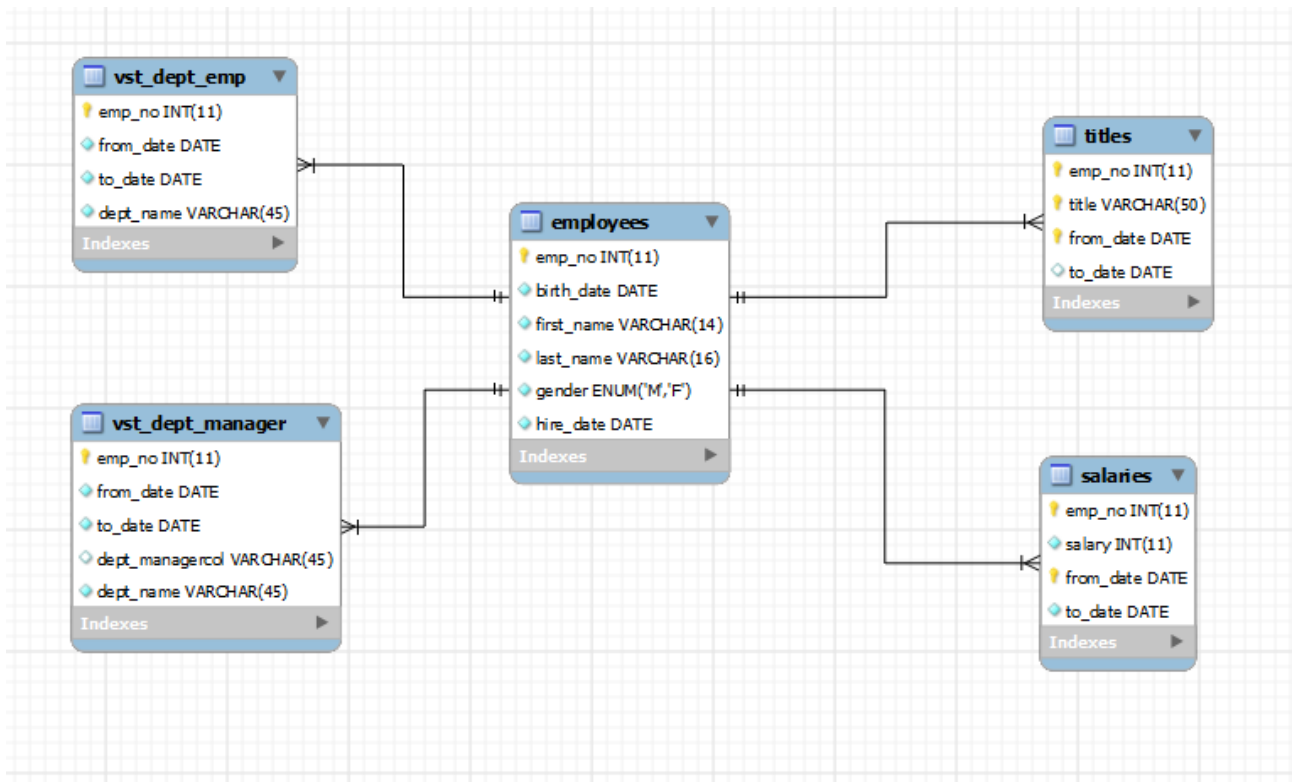
Una fase molt important del projecte es la selecció i creació dels dos models de bases de dades que utilitzarem. Com ja hem comentat abans seran un model d'única taula i un model en estrella.

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.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

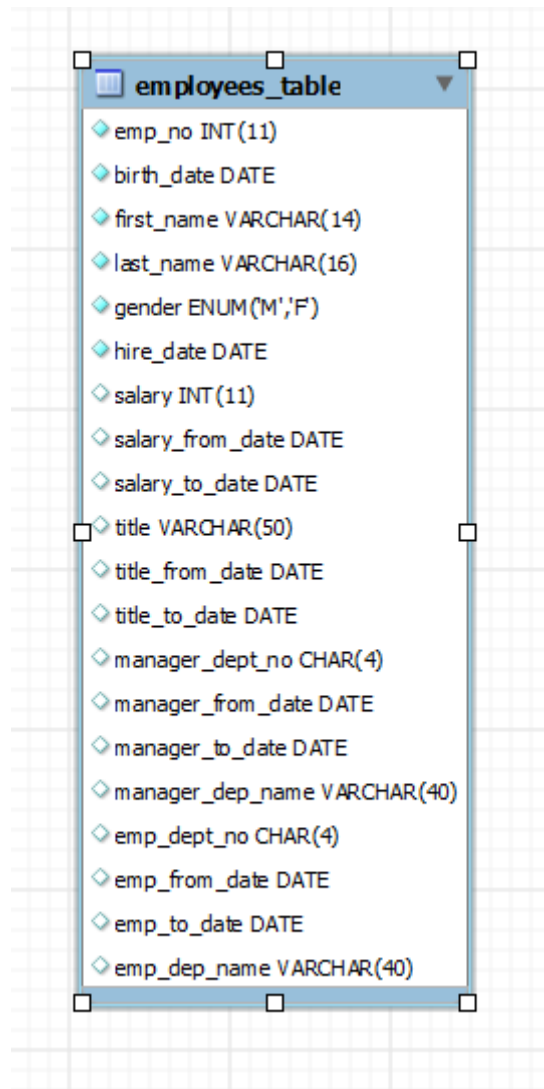
A priori no sembla una en estrella pero conceptualment ho és. En realitat la taula departments es una taula auxiliar de les taules manager i empleat pero per simplificar mitjançant les podriem considerar d'aquesta manera:



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

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

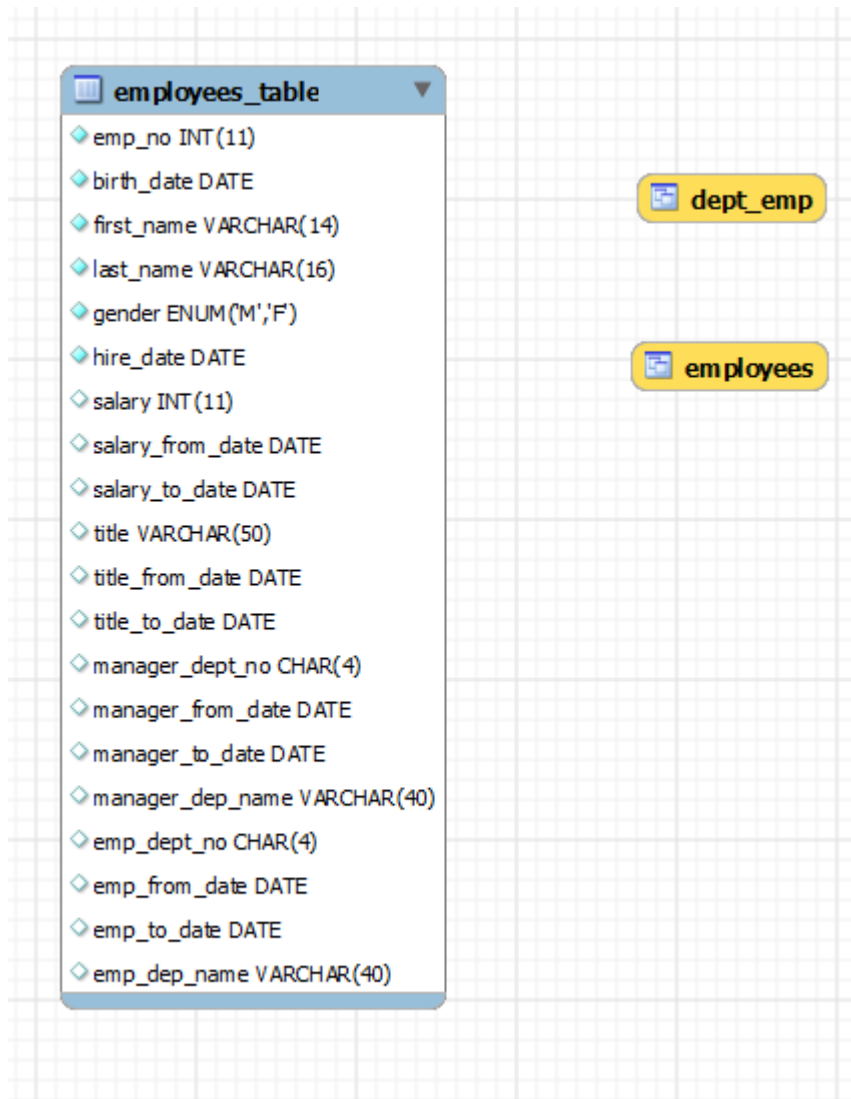
L'altre base de dades amb un tipus de model d'única taula la crearem a partir de la d'estrella a partir d'una consulta que engloba totes les taules fent left outer joins. El resultat d'aquesta consulta la inserirem en una taula molt gran que inclourà totes les dades de l'anterior model.





## Comparativa de rendiment OLAP per a diferents motors de bases de dades

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

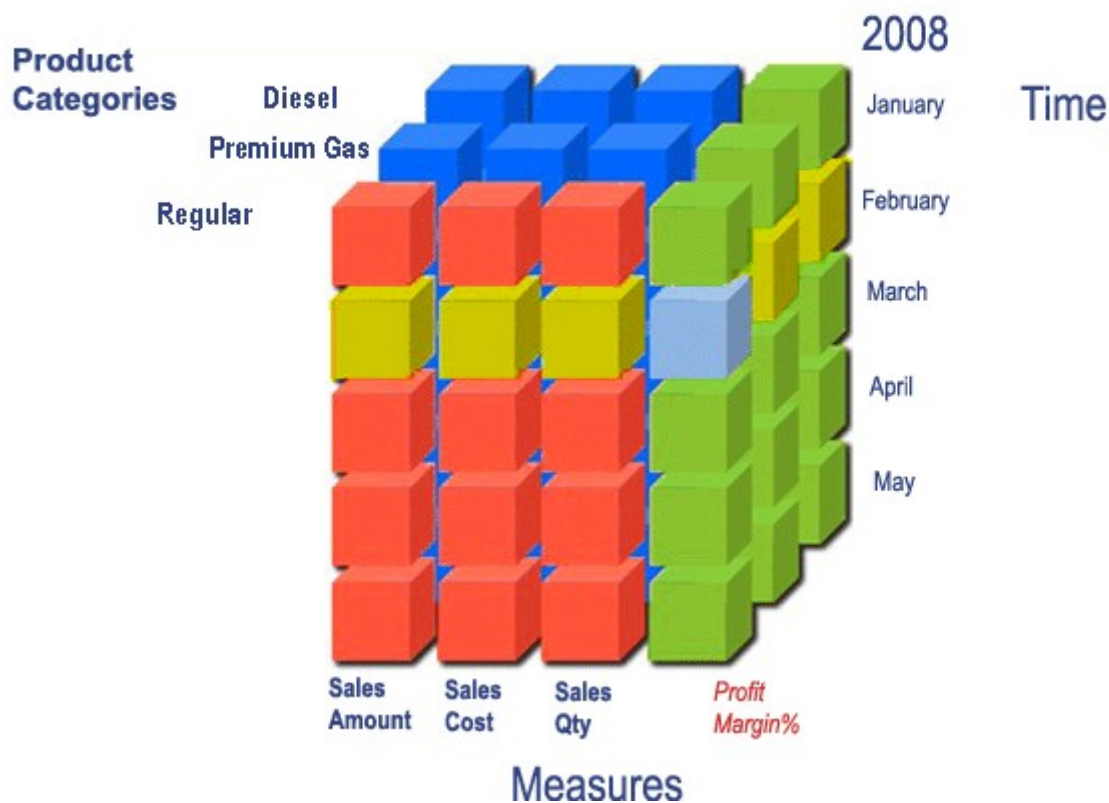


## Comparativa de rendiment OLAP per a diferents motors de bases de dades

### ***Cubs de Dimensions***

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 perquè s'anomena cub olap o cub de dimensions veiem en la següent figura un exemple de cub on tenim la dimensió temps a la dreta, la dimensió 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, perquè el cub està dissenyat amb les mesures i dimensions que ens permeten fer-ho.



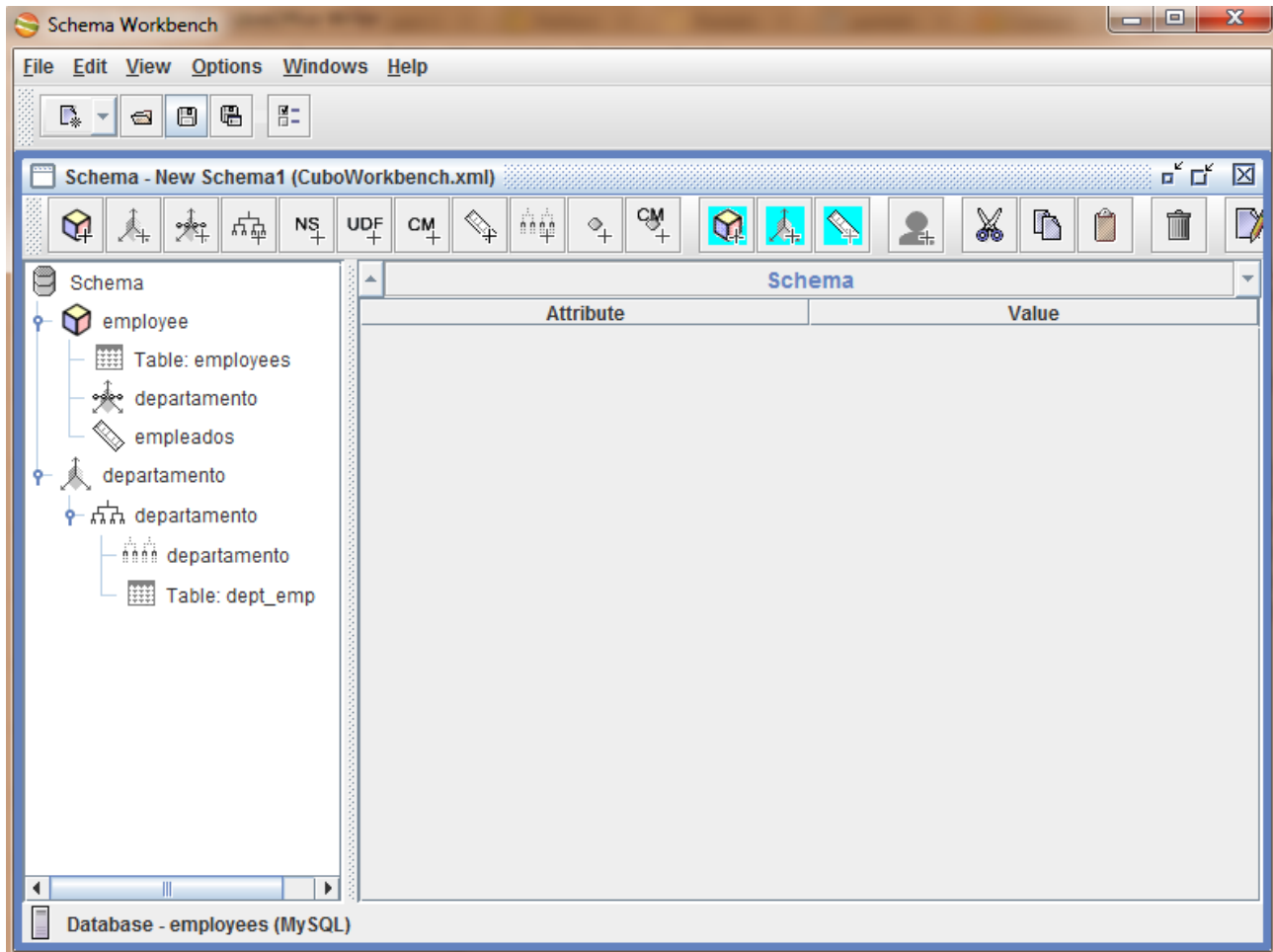
## Comparativa de rendiment OLAP per a diferents motors de bases de dades

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 duem 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.

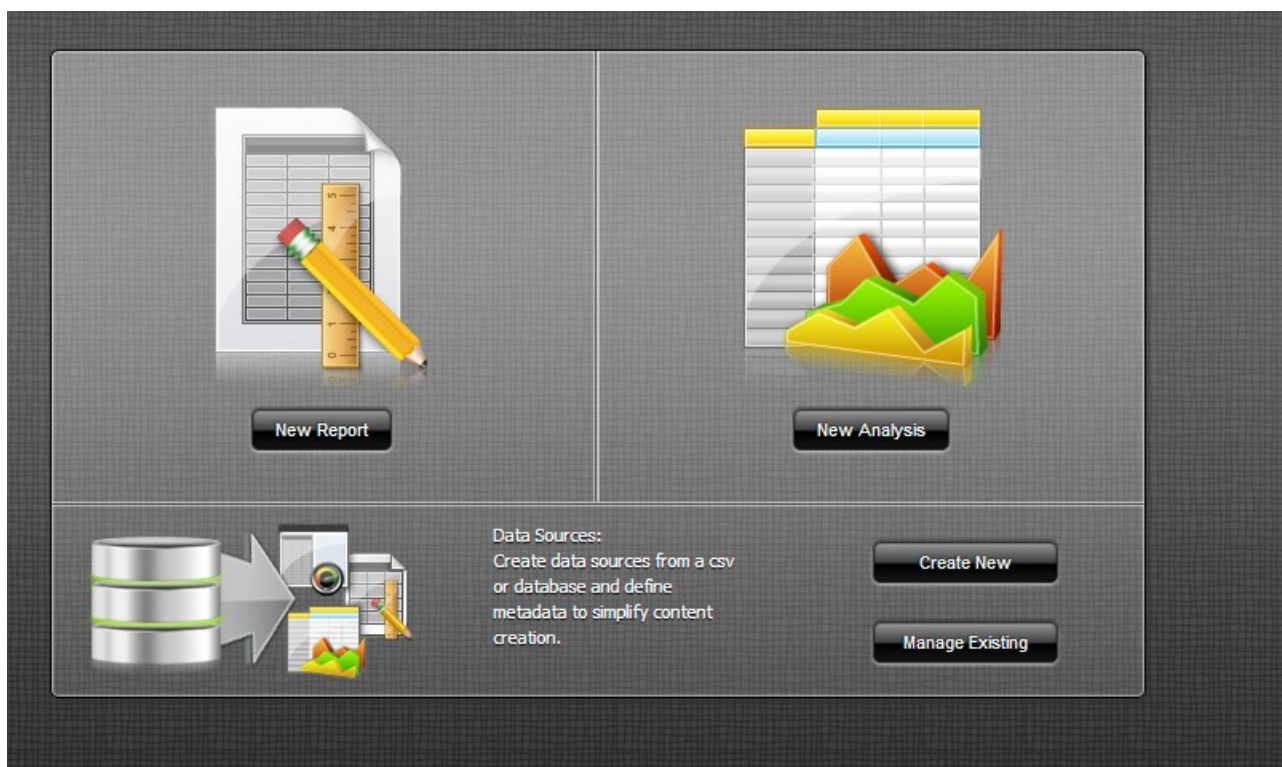
## Comparativa de rendiment OLAP per a diferents motors de bases de dades



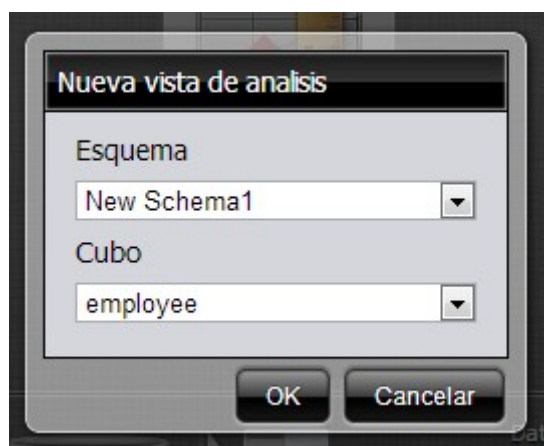
### ***Anàlisi de Dades***

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

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

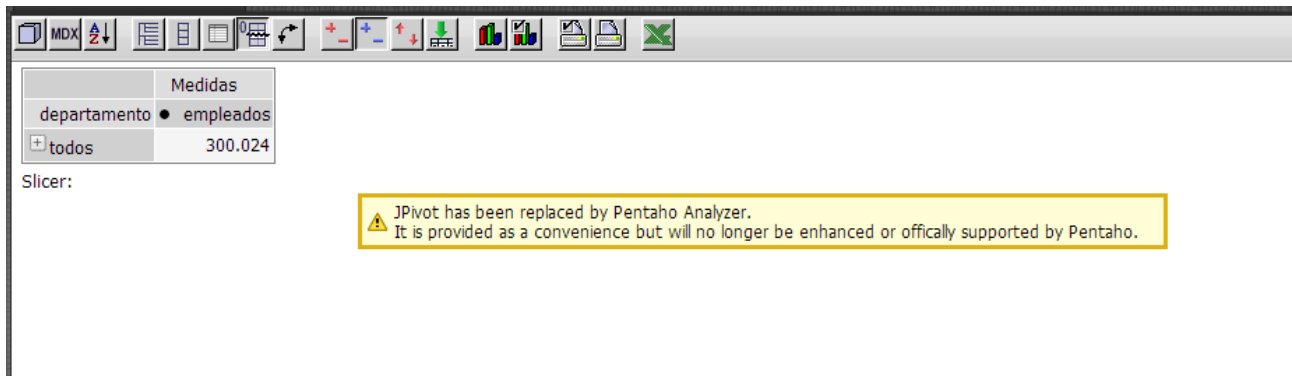


Al fer clic al boto "New Analysis" i seleccionar els cubs que necessitem en aquell moment.



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

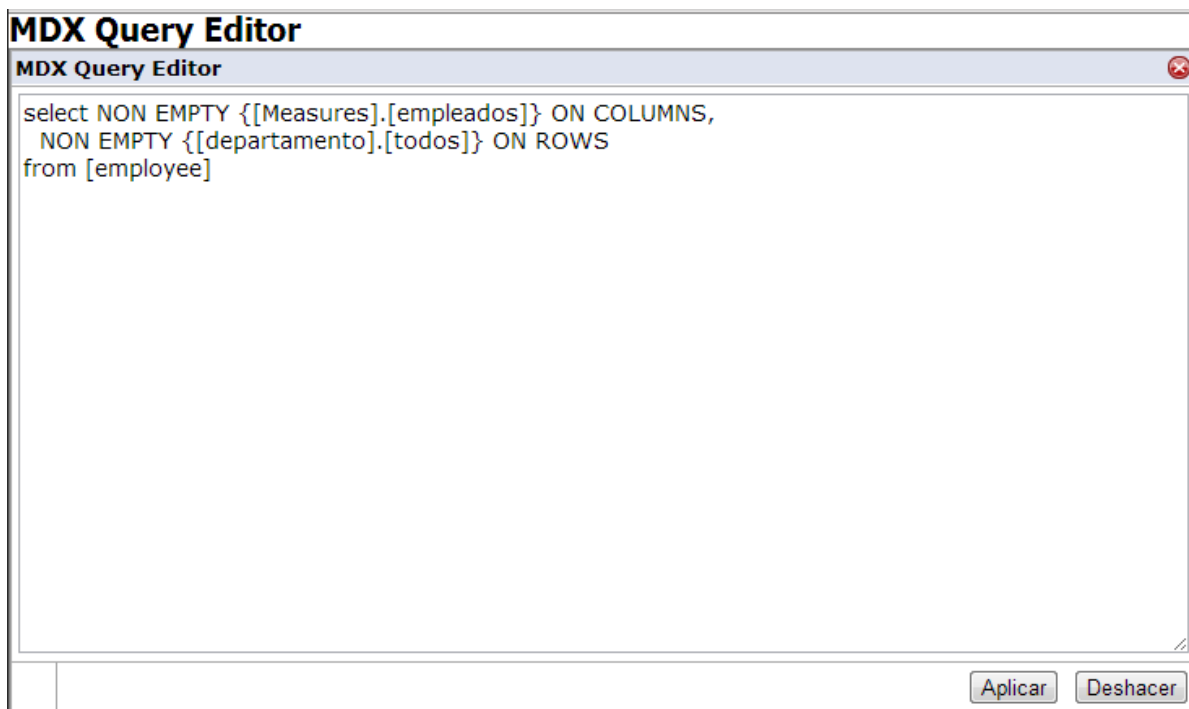
## Comparativa de rendiment OLAP per a diferents motors de bases de dades



Podem redefinir en qualsevol 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:



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

### **Monitorització del temps de resposta**

Abans de poder mesurar els temps de resposta haurem de modificar els paràmetres de configuració del pentaho per que afegixi informació addicional als logs que es generen per defecte. Això ho durem a terme modificant l'arxiu [BI-SERVER FOLDER]/tomcat/webapps/pentaho/WEB-INF/classes/log4j.xml

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.

Un cop localitzades les peticions d'un model i d'un altre realitzarem una comparativa dels temps contra cadascun dels models.

L'arxiu està ubicat a la ruta [BI-SERVER]\tomcat\bin\pentaho.log

```
-----  
2009-12-21 12:29:24,078 DEBUG [mondrian.sql] 16: RolapStar.Column.getCardinal  
from "DIM_TIME" as "DIM_TIME"]  
2009-12-21 12:29:24,078 DEBUG [mondrian.sql] 16: , exec 0 ms  
2009-12-21 12:29:24,078 DEBUG [mondrian.sql] 16: , exec+fetch 0 ms, 1 rows  
2009-12-21 12:29:24,093 DEBUG [mondrian.sql] 17: Segment.load: executing sql  
as "c1", "DIM_TIME"."YEAR_ID" as "c2", sum("ORDERFACT"."TOTALPRICE") as "m0"  
"PRODUCTS" as "PRODUCTS", "DIM_TIME" as "DIM_TIME" where "ORDERFACT"."CUSTOME  
"ORDERFACT"."PRODUCTCODE" = "PRODUCTS"."PRODUCTCODE" and "ORDERFACT"."TIME_ID  
"PRODUCTS"."PRODUCTLINE", "DIM_TIME"."YEAR_ID"]  
2009-12-21 12:29:24,375 DEBUG [mondrian.sql] 17: , exec 266 ms  
2009-12-21 12:29:24,390 DEBUG [mondrian.sql] 17: , exec+fetch 281 ms, 79 rows
```

*Il·lustración 1: Exemple de pentaho.log: font: <http://pentaho-en.phi-integration.com/bi-server/how-to-trace-sql-generated-by-mondrian>*

## RESULTATS

Un cop acabat el desenvolupament de tot el necessari per poder dur a terme les proves de rendiment tenim els següents elements a punt:

Un servidor pentaho bi que ens proveirà d'una infraestructura necessària per compilar solucions de business intelligence. El seu framework ens proveirà de una interfície on hi podrem navegar per els cubs de dades connectats al Mysql a les dues tipologies de bases de dades a testejar.

El pentaho també tindrà configurat el sistema de logs perquè sigui molt explícit en quant a execucions, printant un resum bastant detallat de les execucions de les consultes a MySQL. Això es el que ens donara el indicador principal del rendiment de les consultes contra una base de dades i contra l'altre.

Un Mysql amb dues bases de dades, una en format de taula i l'altre en format d'estrella amb les mateixes dades.

Dos cubs olap configurats amb una dimensió "departamento" apuntant a cada una de les bases de dades per poder fer exactament la mateixa petició.

El guió de la prova sera la següent:

Arrancarem el servidor pentaho amb el seu respectiu mysql i entrarem des de un altre ordinador via web al servidor: El servidor a la seva arrencada obre un servidor tomcat que es queda escoltant per el port 8080, així per entrar des de un altre ordinador de la mateixa xarxa posarem: "[http://ip\\_del\\_server:8080/](http://ip_del_server:8080/)" això ens obrira el portal del pentaho bi server amb la seva pantalla de login



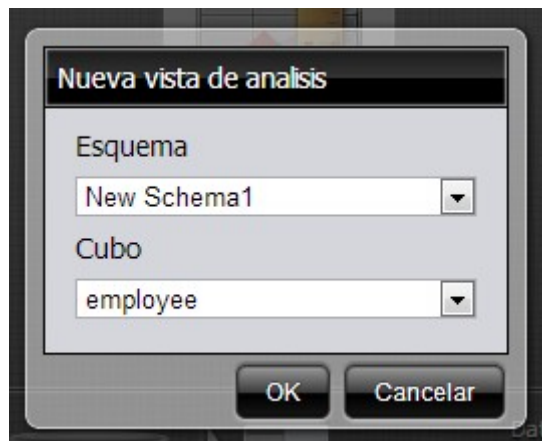
## Comparativa de rendiment OLAP per a diferents motors de bases de dades

Entrarem les credencials necessàries (pentaho per defecte porta uns usuaris de prova amb drets de administració que utilitzarem per dur a terme la prova).

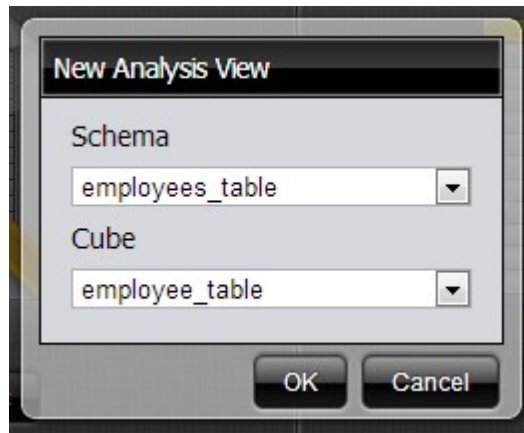
Això ens porta al workspace del usuari loguejat on tenim un munt d'opcions que ens ofereix Pentaho totes relacionades amb el anàlisis de dades. Aquesta pàgina es totalment reconfigurable per el usuari i no té per que ser així, ja que pentaho brinda la possibilitat de configurar-la programar-la i fins i tot instal·lar-hi eines addicionals.

Nosaltres només necessitarem crear un nou anàlisis de dades per poder accedir al Jpivot del nostre cub.

Primer accedirem amb el cub del model estrella i després amb el cub del model taula, així tindrem les consultes en el log llestes per poder-les analitzar de manera correcta.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades



Un cop executats els dos cubs ens descarregarem per ftp el log del pentaho i l'analitzarem exhaustivament.

```
2013-06-01 21:32:13,673 DEBUG [mondrian.sql] 0: Segment.load: executing sql [
select
  count(`employees`.`emp_no`) as `m0`
from
  `employees` as `employees`]
2013-06-01 21:32:13,820 DEBUG [mondrian.sql] 0: , exec 145 ms
2013-06-01 21:32:13,831 DEBUG [mondrian.sql] 0: , exec+fetch 158 ms, 1 rows
2013-06-01 21:32:16,616 DEBUG [mondrian.sql] 1:
SqlMemberSource.getMemberChildren: executing sql [
select
  `dept_emp`.`dept_no` as `c0`
from
  `dept_emp` as `dept_emp`
group by
  `dept_emp`.`dept_no`
order by
```

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

```
ISNULL(`dept_emp`.`dept_no`) ASC, `dept_emp`.`dept_no` ASC]
2013-06-01 21:32:16,618 DEBUG [mondrian.sql] 1: , exec 0 ms
2013-06-01 21:32:16,619 DEBUG [mondrian.sql] 1: , exec+fetch 2 ms, 9 rows
2013-06-01 21:32:16,627 DEBUG [mondrian.sql] 2: RolapStar.Column.getCardinality:
executing sql [
select
  count(distinct `dept_emp`.`dept_no`) as `c0`
from
  `dept_emp` as `dept_emp`]
2013-06-01 21:32:16,778 DEBUG [mondrian.sql] 2: , exec 150 ms
2013-06-01 21:32:16,778 DEBUG [mondrian.sql] 2: , exec+fetch 151 ms, 1 rows

2013-06-01 21:47:58,698 DEBUG [mondrian.sql] 4: Segment.load: executing sql [
select
  count(`employees`.`emp_no`) as `m0`
from
  `employees` as `employees`]
2013-06-01 21:48:10,275 DEBUG [mondrian.sql] 4: , exec 11576 ms
2013-06-01 21:48:10,277 DEBUG [mondrian.sql] 4: , exec+fetch 11578 ms, 1 rows
2013-06-01 21:48:12,555 DEBUG [mondrian.sql] 5:
SqlMemberSource.getMemberChildren: executing sql [
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]
```

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

```
2013-06-01 21:48:18,940 DEBUG [mondrian.sql] 5: , exec 6384 ms
2013-06-01 21:48:18,941 DEBUG [mondrian.sql] 5: , exec+fetch 6386 ms, 9 rows
2013-06-01 21:48:18,942 DEBUG [mondrian.sql] 6: RolapStar.Column.getCardinality:
executing sql [
select
  count(distinct `dept_emp`.`dept_no`) as `c0`
from
  `dept_emp` as `dept_emp`]
2013-06-01 21:48:25,180 DEBUG [mondrian.sql] 6: , exec 6238 ms
2013-06-01 21:48:25,181 DEBUG [mondrian.sql] 6: , exec+fetch 6239 ms, 1 rows
```

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.

## CONCLUSIONS

### *Objectius aconseguits*

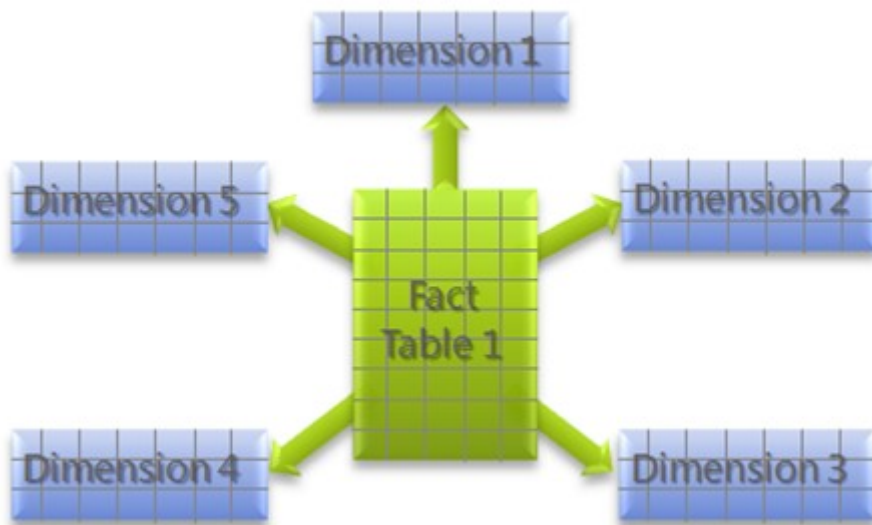
A partir dels resultats aconseguits m'ha permès distingir si hi ha algun canvi en el rendiment d'un o del altre model. Aquest era el principal objectiu del projecte i a continuació dono les conclusions.

Com hem vist en l'apartat anterior podem confirmar que el model en estrella és molt més ràpid que el model en taula, però per què?

Hi ha varies raons per les quals un model en estrella té un millor rendiment que un model en taula:

Primerament el model de cubs dimensionals com ja hem comentat requereix d'una taula de fets i una de dimensions.

## Comparativa de rendiment OLAP per a diferents motors de bases de dades



Dit d'una altra manera necessitem una taula que compta i una taula que classifica les dades. En aquest cas la taula de fets es la empleats i la taula de dimensions es el departament que pertany. D'aquesta manera hem obtingut un cub que calcula els empleats que hi ha per cada departament.

Aquest tipus de model necessita que al Mysql això també estigui representat. En el model d'estrella això ho obtenim de manera natural (encara que a vegades no té per que ser així i haguem de fer alguna transformació o haguem de ajudar-nos amb vistes, com és el cas del model amb taules), però no en el model de taula.

Aquí roman una de les deficiències més importants d'aquest model alhora d'utilitzar-lo per l'anàlisi de cubs dimensionals, ja que hem de crear vistes que mostrin les dades de la taula en un format que permeti després utilitzar-les en les nostres mesures i dimensions.

Les vistes en definitiva només fan que afegir un pas intermedi adicional que ralentitza les peticions.

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

A més a més aquestes vistes operen contra una taula que conté tota la informació de la base de dades, per tant, la vista que conté els departaments els obté de fer una consulta contra els milers de registres que conté la taula, en comparació amb el model d'estrella que ja té una taula amb els tipus de departaments que hi ha.

Un cop dit això, podem constatar amb tota seguretat que la diferència més important és aquesta: un model estrella permet operar amb les dades justes que necessitem en aquell moment, en canvi un model en taula sempre estarem consultant contra tota la informació de la base de dades.

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

### ***Experiència Personal***

Anteriorment havia treballat en una consultoria de Màrketig mentres que paral·lelament cursava el màster de programari lliure. Vaig tenir l'oportunitat de posar en pràctica els coneixements en aquesta empresa, com a resultat vaig aconseguir que l'empresa migrés gran part dels seus sistemes a programari lliure: servidors debien amb mysql i apache que allotjaven aplicacions programades per mi en php que s'encarregaven de recolectar dades d'enquestes sobretot. Vam introduir el pentaho com a motor d'anàlisi de dades, tot i que la part de disseny i creació de cubs no la feia jo.

Gran part dels coneixements MySql i administració del servidor utilitzada en aquest projecte les vaig aprendre en les assignatures d'administració de sistemes gnu linux, Bases de dades, etc... i van ser posades en pràctica en aquesta empresa que vaig poder utilitzar exitosament com a banc de proves.

Tenia una idea general de com funcionava el pentaho bi server però mai m'havia encarregat personalment de la posada a punt d'un servidor, amb tot el que això comporta, ni m'havia ficat a programar els cubs de dimensions amb el workbench ja que necessitava d'uns coneixements d'analista de dades que no tinc.

La instal·lació del Pentaho i el Mysql en una màquina debien en definitiva m'han servit per assentar coneixements i poder fer-ho sense l'ajuda de ningú també m'ha ajudat a solucionar problemes per mi sol, cosa que en la consultoria on vaig estar segurament ho hauria fet amb mes companys i no hagués sigut tant realitzador per mi.



## ***Aprentatge***

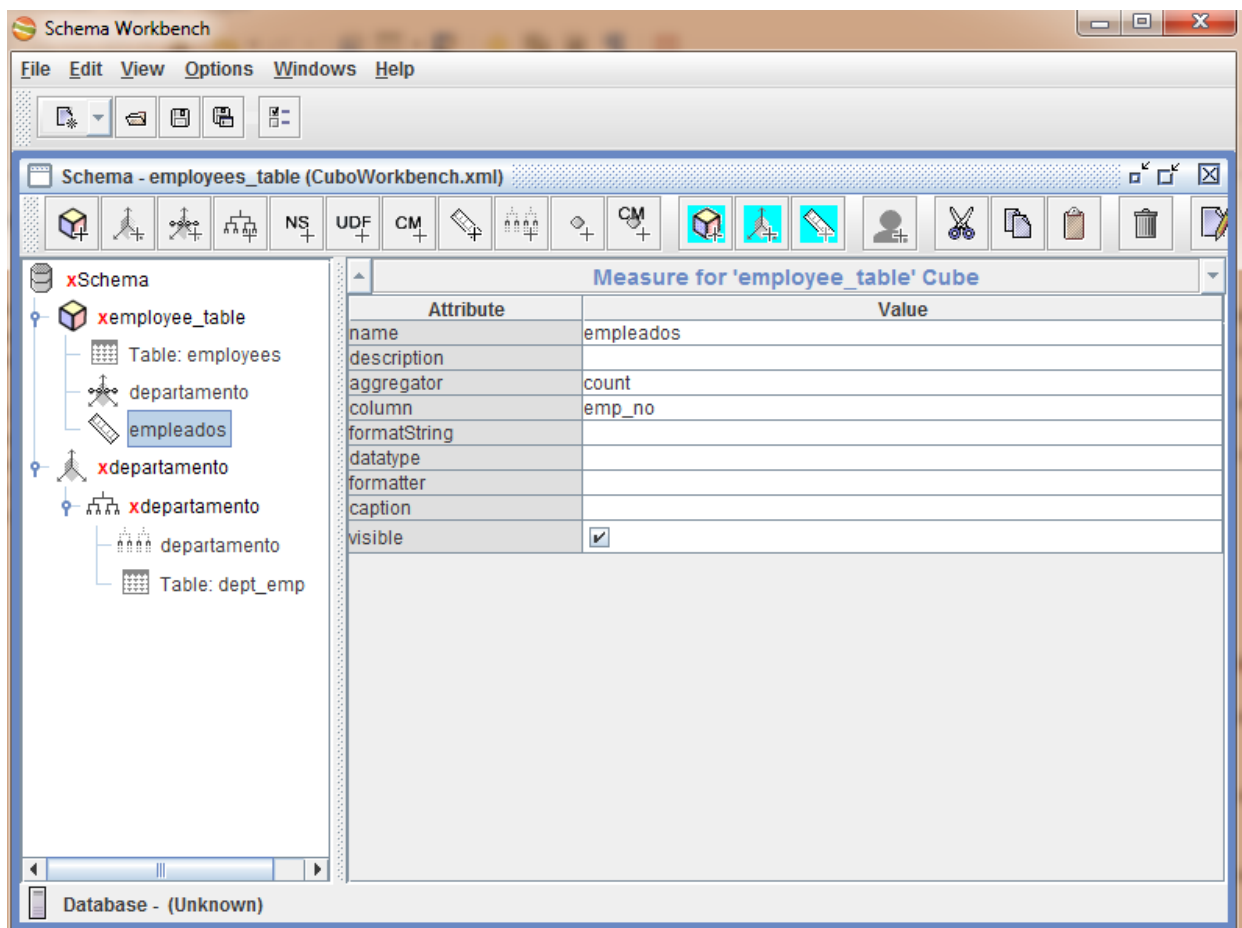
El més important d'aquest projecte ha estat poder desenvolupar un sistema com el del projecte amb la guia dels consultors obviament i única i exclusivament amb eines de programari lliure. Ha sigut una síntesis de molts coneixements apresos durant el màster. En el transcurs d'aquest projecte he trobat molts esculls que m'han permès desenvolupar uns coneixements nous molt interessants:

- **Pentaho Bi Server:** La instal·lació i posada a punt d'un servidor pentaho era una cosa que en gran part desconeixia i he après gràcies a aquest projecte. Va costar bastants intents arribar a tenir una màquina que estigués llesta per poder utilitzar-se per aquest projecte ja que el escull mes important va ser el de que el servidor tingués com a origen de dades per defecte el servidor mysql allotjat al mateix localhost. S'havien de modificar una serie d'arxius en punts determinats del arbre de directoris del pentaho i si no es feia correctament la màquina no arrancava correctament i era molt difícil detectar on estava el problema.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

- **Pentaho schema workbench:** Aquesta eina es la que mes alegries m'ha aportat en aquest projecte. Es podria dir que gran part del meu interès en dur a terme aquest projecte era poder dissenyar els cubs dimensionals amb aquesta eina encara que els cubs no fossin massa complexes, lo necessariament complexes que el projecte requeria. Era una cosa que havia vist fer a on treballava però com que el disseny daquests cubs no era de la meva competència mai vaig poder aprendre a fer-ho.



## Comparativa de rendiment OLAP per a diferents motors de bases de dades

- **Utilització de logs i existència d'altres eines complementàries al pentaho:** La monitorització del servidor també ha sigut una de les parts que m'ha fet descobrir un altre ventall de possibilitats no només en el seguiment de les consultes de mysql sinó també en la detecció d'errors en general, o en la informació dels comportaments interns del servidor.

Abans he comentat que el pentaho bi server té moltes altres eines que poden ser instal·lades a posteriori com per exemple el saiku o les eines CDF que comentarem mes endavant. Tot això i més pot ser configurat perquè deixi empremta en aquests logs i així poder fer un seguiment del funcionament ,no només del nucli del pentaho sinó també de les eines externes que poden utilitzar-se a posteriori.

```
162 <!-- ===== -->
163 <!-- Special Log File specifically for Mondrian SQL Statements -->
164 <!-- ===== -->
165
166
167 <appender name="SQLLOG" class="org.apache.log4j.RollingFileAppender">
168     <param name="File" value="mondrian_sql.log"/>
169     <param name="Append" value="false"/>
170     <param name="MaxFileSize" value="500KB"/>
171     <param name="MaxBackupIndex" value="1"/>
172     <layout class="org.apache.log4j.PatternLayout">
173         <param name="ConversionPattern" value="%d %-5p [%c] %m%n"/>
174     </layout>
175 </appender>
176
177 <category name="mondrian.sql">
178     <priority value="DEBUG"/>
179     <appender-ref ref="SQLLOG"/>
180 </category>
181
182 </log4j:configuration>
```

## Futures aplicacions i vies de continuació

La veritat es que la situació un cop arribat a aquest punt es bastant inicial, el que vull dir amb això es que tot just acaba de començar. He obtingut els coneixements suficients com pera administrar un servidor pentaho i saber utilitzar-lo en les seves funcions més bàsiques fins el punt d'arribar a discernir entre quin dels models de taules era mes eficient. Ara la via principal de continuació es explotar aquest motor pentaho i totes les seves eines complementaries.

Anem a fer un repàs una mica exhaustiu d'aquestes:

- **Saiku:**   
CUTTING EDGE OPEN SOURCE ANALYTICS

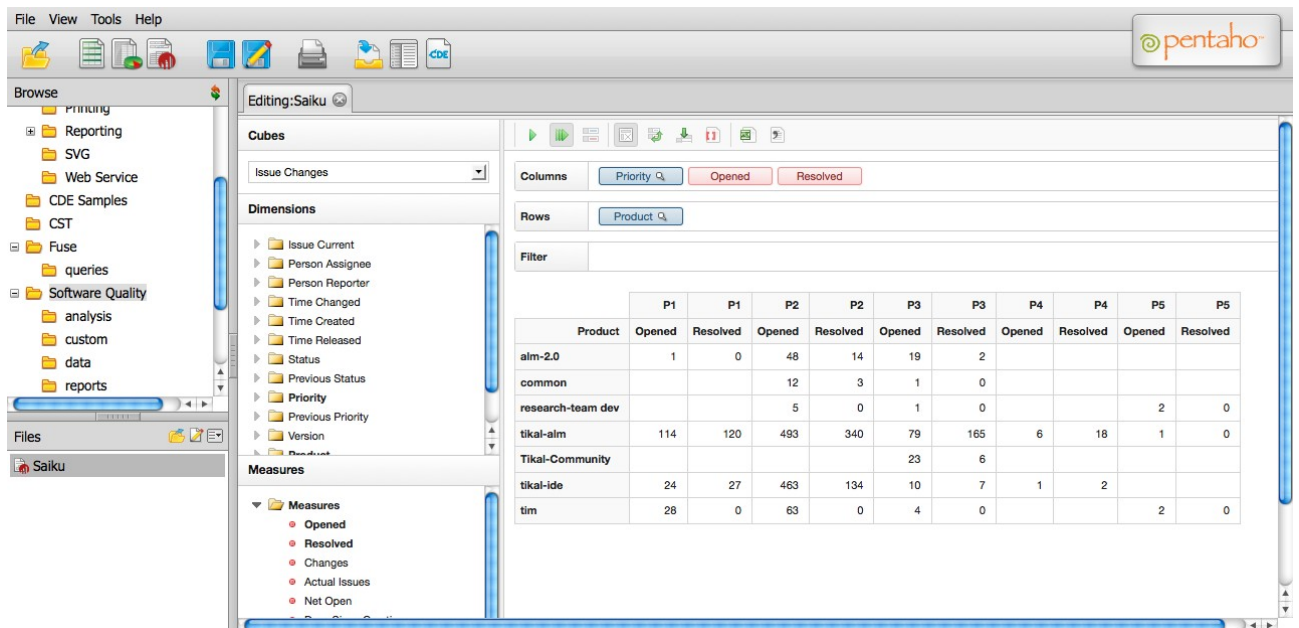
Saiku es una eina de software lliure totalment modular, d'anàlisi de OLAP lleuger i fàcilment incrustable extensible i configurable.

Consisteix en un servidor basat en REST que es conecta a un servidor OLAP (en aquest cas el pentaho) i mitjançant una interfície molt intuïtiva basada en JQuery ens permet manegar les mesures i dimensions com feiem anteriorment amb el Jpivot.

Aquesta alternativa pretén ser una continuació del Jpivot existent ja que aquest s'està quedant obsolet.

A continuació una captura de pantalla amb la interfície del saiku.

## Comparativa de rendiment OLAP per a diferents motors de bases de dades



### CommunityDashboardFramework

- CDF Community Dashboard Framework:



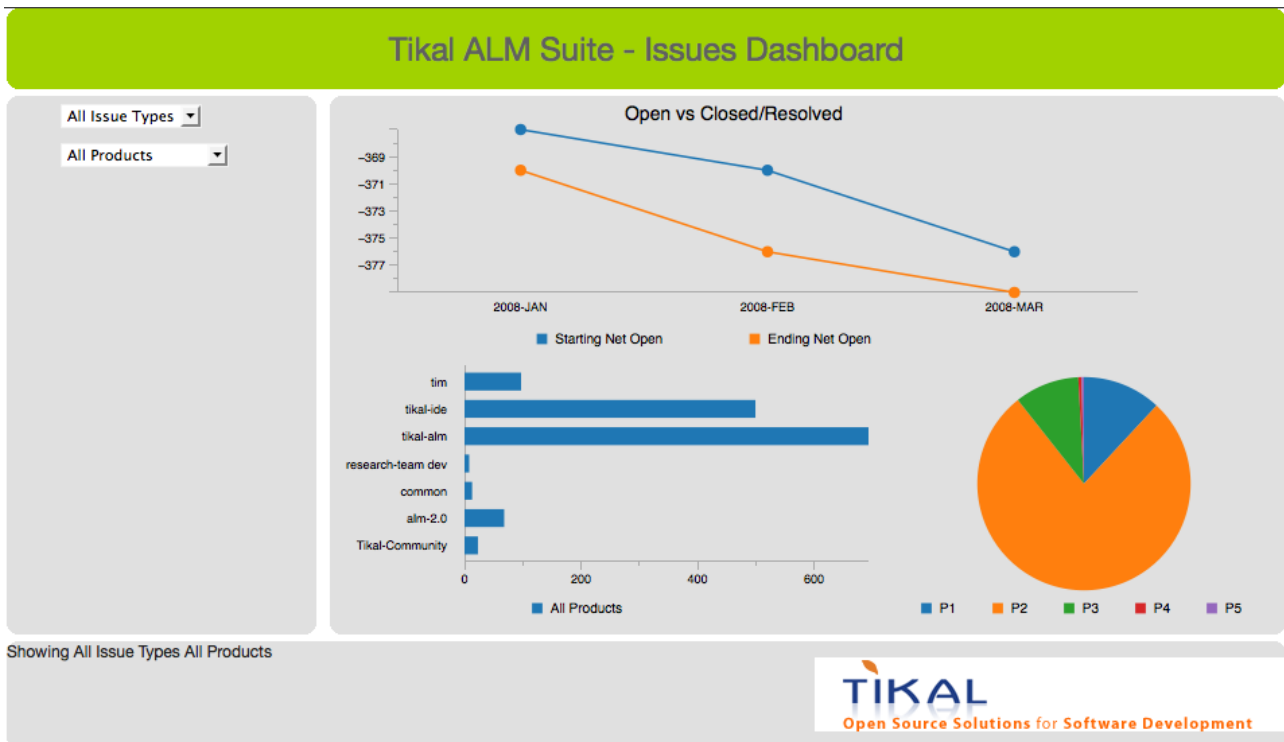
L'objectiu d'aquest framework es principalment la integració de gràfics atractius al àmbit del pentaho biserver.

El projecte, iniciat per Pedro Alves i Ingo Klose, es basa en el framework d'Ingo i afegeix algunes noves característiques, mantenint una compatibilitat amb el marc inicial. Algunes de les característiques mes importants són:

- Generació automàtica d'objectes de formulari, com quadres de text i selectors de data.
- Separació del disseny html i definició del component.
- Integració OpenStreetMaps.
- Portlet d'integració.

## Comparativa de rendiment OLAP per a diferents motors de bases de dades

Aquesta primera versió és només el primer pas en el viatge a l'usuari i el dissenyador de gràfics usables, perfectament integrats a la suite de BI de Pentaho.



# Comparativa de rendiment OLAP per a diferents motors de bases de dades

- **Kettle**

El pentaho Data integration comuniti edition (PDI CE) també conegut com a kettle, integra una solució destinada a l'extracció, transformació i càrrega (Extraction Transformation and Loading ETL). Amb una interfície gràfica i intuïtiva ofereix la possibilitat de arrossegar les transformacions i els processos i entrellaçar-los de manera senzilla. És community edition per que es la versió open source, però també disposa d'una versio enterprise. El client que programa tot això s'anomena spoon:

To test this transformation, you will need to:

- Make sure the Hypersonic sample database is running (`!pdi-ee\data-integration-server\data/start_hypersonic.bat`)
- Open the Table Output step and click the SQL button to create the target output table

Stepname	Copynr	Read	Written	Input	Output	Updated	Rejected	Errors	Active	Time	Speed (r/s)	inp
1 Filter Missing Zips	0	2823	2823	0	0	0	0	0	Finished	0.5	6019.1	
2 Lookup Missing Zips	0	21455	76	0	0	0	0	0	Finished	0.9	24520.0	
3 Read Postal Codes	0	0	21379	21380	0	1	0	0	Finished	0.7	31815.4	
4 Prepare Field Layout	0	76	76	0	0	0	0	0	Finished	0.9	85.2	
5 Value Mapper	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4	
6 Read Sales Data	0	0	2823	2824	0	1	0	0	Finished	0.3	8209.3	
7 Select values	0	2823	2823	0	0	0	0	0	Finished	0.9	3112.4	
8 Number range	0	2823	2823	0	0	0	0	0	Finished	0.9	3061.8	
9 Write to Database	0	2823	2823	0	2823	0	0	0	Finished	1.1	2543.2	

## BIBLIOGRAFÍA

Guió d'instal·lació i configuració del pentaho bi server

<http://interestingittips.wordpress.com/2011/04/27/complete-pentaho-installation-on-ubuntu-part-1/>

Com publicar un cub utilitzant l'esquema workbench

<http://wiki.pentaho.com/display/ServerDoc1x/Publishing+an+Analysis+Schema+Using+Schema+Workbench>

Configurar els orígens de dades

<http://wiki.pentaho.com/display/ServerDoc1x/Configuring+JNDI+Datasources+for+use+in+the+Pentaho+BI+Platform>

Com monitoritzar les consultes que fa el pentaho a mysql

<http://pentaho-en.phi-integration.com/bi-server/how-to-trace-sql-generated-by-mondrian>