



# APLICACION DE BASES DE DATOS

Informe de prácticas

Practica JPA

Alumno:

Juan Luis García González

Alejandro Ortega Martínez

Mayo 2021

## Contenido

Preparación del Informe .....	3
Método 1 – Insertar Incidencia .....	3
Método 2 – Indultar .....	5
Método 3 – Insertar nuevo tipo de incidencia .....	6
Método 4 – Consultar Numero de conductores con incidencias en asociación. ....	6
Método 5 – Consultar Ranking .....	7
Método 6 – Consultar Asociaciones .....	8
Consideraciones .....	8
Bibliografía .....	9

## Preparación del Informe

Para la obtención de las consultas, deberemos cambiar previamente unos parámetros en el archivo “persistence.xml”:

```
<properties>

<property name="hibernate.dialect" value="org.hibernate.dialect.Oracle10gDialect" />
<property name="hibernate.connection.driver_class" value="oracle.jdbc.driver.OracleDriver" />
<property name="hibernate.connection.url" value="jdbc:oracle:thin:@localhost:1521:XE" />
<property name="hibernate.default_schema" value="HR" />
<property name="hibernate.connection.username" value="HR" />
<property name="hibernate.connection.password" value="hr" />
<property name="hibernate.connection.autocommit" value="false" />
<property name="hibernate.show_sql" value="true" />
<property name="hibernate.format_sql" value="true" />
<property name="hibernate.generate_statistics" value="true" />
<property name="hibernate.use_sql_comments" value="true" />


```

De esta forma generaremos , las distintas estadísticas asociadas a cada consulta, mostraremos las consultas y su formato.

## Método 1 – Insertar Incidencia

El método insertar incidencia, a nivel de consulta, se inicia con la búsqueda del conductor con el nif que nos pasan por parámetros. A nivel de código se corresponde con un findby :

```
Conductor conductor = conductorDAO.findById(nif);
```

Hibernate realiza la siguiente consulta asociada:

```
select
  conductor0_.nif as nif1_2_0_,
  conductor0_.apellido as apellido2_2_0_,
  conductor0_.direccion as direccion3_2_0_,
  conductor0_.ciudad as ciudad4_2_0_,
  conductor0_.CP as cp5_2_0_,
  conductor0_.nombre as nombre6_2_0_,
  conductor0_.puntos as puntos7_2_0_
from
  HR.Conductor conductor0_
where
  conductor0_.nif=?
```

Podemos apreciar que simplemente carga los datos del conductor, solo guarda los valores relativos a la clase ignorando las relaciones. Esto se debe a que las relaciones son ""toMany. Haciendo carga por defecto LAZY.

El siguiente paso y de forma análoga, buscamos el tipo de inidencia:

```
TipoIncidencia tipoIncidencia = tipoIncidenciaDAO.findById(tipo);
```

Hibernate realiza la siguiente consulta:

```

Hibernate:
select
    tipoincide0_.id as id1_4_0_,
    tipoincide0_.descripcion as descripcion2_4_0_,
    tipoincide0_.valor as valor3_4_0_
from
    HR.TipoIncidencia tipoincide0_
where
    tipoincide0_.id=?

```

Al añadir la incidencia :

```
conductor.addIncidencia(incidencia);
```

Debe de cargar el conjunto de incidencias que, como se ha comentado, tiene carga LAZY (De ahí que se cargue solo cuando se referencia):

```

Hibernate:
select
    incidencia0_.nif as nif2_3_0_,
    incidencia0_.fecha as fecha1_3_0_,
    incidencia0_.fecha as fecha1_3_1_,
    incidencia0_.nif as nif2_3_1_,
    incidencia0_.anotacion as anotacion3_3_1_,
    incidencia0_.idtipo as idtipo4_3_1_,
    tipoincidel1_.id as id1_4_2_,
    tipoincidel1_.descripcion as descripcion2_4_2_,
    tipoincidel1_.valor as valor3_4_2_
from
    HR.Incidencia incidencia0_
left outer join
    HR.TipoIncidencia tipoincidel1_
        on incidencia0_.idtipo=tipoincidel1_.id
where
    incidencia0_.nif=?

```

Cabe destacar que la relación de incidencia y tipo de incidencia desde incidencia es ""toOne , cargándose por defecto también tipo de incidencia.

Las dos ultimas consultas, corresponde a la actuación de persistir la nueva incidencia creada, y de actualizar los puntos del conductor:

```

Hibernate:
/* insert es.ubu.lsi.model.asociacion.Incidencia */ insert
into
    HR.Incidencia
    (anotacion, idtipo, fecha, nif)
values
    (?, ?, ?, ?)
Hibernate:
/* update
es.ubu.lsi.model.asociacion.Conductor */ update
HR.Conductor
set
    apellido=?,
    direccion=?,
    ciudad=?,
    CP=?,
    nombre=?,
    puntos=?
where
    nif=?

```

## Método 2 – Indultar

De la misma forma que el método anterior, deberemos encontrar al conductor que nos pasan por parámetro. Para ello, utilizaremos el `findById` cuya consulta interna es la siguiente:

```

Hibernate:
  select
    conductor0_.nif as nif1_2_0_,
    conductor0_.apellido as apellido2_2_0_,
    conductor0_.direccion as direccion3_2_0_,
    conductor0_.ciudad as ciudad4_2_0_,
    conductor0_.CP as cp5_2_0_,
    conductor0_.nombre as nombre6_2_0_,
    conductor0_.puntos as puntos7_2_0_
  from
    HR.Conductor conductor0_
  where
    conductor0_.nif=?

```

Basándonos en los comentarios realizados anteriormente, para realizar el borrado de las incidencias, deberemos primero cargarlas haciendo una nueva consulta:

```

Hibernate:
  select
    incidencia0_.nif as nif2_3_0_,
    incidencia0_.fecha as fecha1_3_0_,
    incidencia0_.fecha as fecha1_3_1_,
    incidencia0_.nif as nif2_3_1_,
    incidencia0_.anotacion as anotacion3_3_1_,
    incidencia0_.idtipo as idtipo4_3_1_,
    tipoincid1_.id as id1_4_2_,
    tipoincid1_.descripcion as descripcion2_4_2_,
    tipoincid1_.valor as valor3_4_2_
  from
    HR.Incidencia incidencia0_
  left outer join
    HR.TipoIncidencia tipoincid1_
    on incidencia0_.idtipo=tipoincid1_.id
  where
    incidencia0_.nif=?

```

Al hacer el commit, se realiza la actualización de los puntos del conductor, y se borran las incidencias. Es de obligada mención, comentar que los borrados se realizan uno a uno por cada incidencia existente (No mediante la misma consulta).

```

Hibernate:
  /* update
    es.ubu.lsi.model.asociacion.Conductor */ update
    HR.Conductor
  set
    apellido=?,
    direccion=?,
    ciudad=?,
    CP=?,
    nombre=?,
    puntos=?
  where
    nif=?

```

```

Hibernate:
/* delete es.ubu.lsi.model.asociacion.Incidencia */ delete
from
  HR.Incidencia
where
  fecha=?
  and nif=?
Hibernate:
/* delete es.ubu.lsi.model.asociacion.Incidencia */ delete
from
  HR.Incidencia
where
  fecha=?
  and nif=?

```

### Método 3 – Insertar nuevo tipo de incidencia

El numero realizado de consultas es muy reducido . Se realiza una para recoger cual es el siguiente numero de id a insertar (Se accede a la tabla secuencia). La otra consulta se basa en hacer el insert :

```

Hibernate:
select
  HR.TIPOINCIDENCIA_SEQ.nextval
from
  dual
Hibernate:
/* insert es.ubu.lsi.model.asociacion.TipoIncidencia
*/ insert
into
  HR.TipoIncidencia
  (descripcion, valor, id)
values
  (?, ?, ?)

```

No hay cabida para más comentarios.

### Método 4 – Consultar Numero de conductores con incidencias en asociación.

Los primero que hacemos es buscar la asociación con el id pasado por parámetro:

```

Hibernate:
select
  asociacion0_.idasoc as idasoc1_0_0_,
  asociacion0_.direccion as direccion2_0_0_,
  asociacion0_.ciudad as ciudad3_0_0_,
  asociacion0_.CP as cp4_0_0_,
  asociacion0_.nombre as nombre5_0_0_
from
  HR.Asociacion asociacion0_
where
  asociacion0_.idasoc=?

```

El siguiente paso es cargar los conductores y sus incidencias. Esto se va relajando en varias consultas, ya que , las asociaciones son ""ToMany (Dando lugar al problema N+1:

```

Hibernate:
  select
    conductore0_.idasoc as idasoc1_1_0_,
    conductore0_.nif as nif2_1_0_,
    conductor1_.nif as nif1_2_1_,
    conductor1_.apellido as apellido2_2_1_,
    conductor1_.direccion as direccion3_2_1_,
    conductor1_.ciudad as ciudad4_2_1_,
    conductor1_.CP as cp5_2_1_,
    conductor1_.nombre as nombre6_2_1_,
    conductor1_.puntos as puntos7_2_1_
  from
    HR.Asociacion_Conductor conductore0_
  inner join
    HR.Conductor conductor1_
    on conductore0_.nif=conductor1_.nif
  where
    conductore0_.idasoc=?
Hibernate:
  select
    incidencia0_.nif as nif2_3_0_,
    incidencia0_.fecha as fecha1_3_0_,
    incidencia0_.fecha as fecha1_3_1_,
    incidencia0_.nif as nif2_3_1_,
    incidencia0_.anotacion as anotacion3_3_1_,
    incidencia0_.idtipo as idtipo4_3_1_,
    tipoincidel_.id as id1_4_2_,
    tipoincidel_.descripcion as descripcion2_4_2_,
    tipoincidel_.valor as valor3_4_2_
  from
    HR.Incidencia incidencia0_
  left outer join
    HR.TipoIncidencia tipoincidel_
    on incidencia0_.idtipo=tipoincidel_.id
  where
    incidencia0_.nif=?
Hibernate:

```

De esta forma, se realizaría con cada conductor de la asociación .

## Método 5 – Consultar Ranking

En este caso , la única consulta que se hace es la que hacemos nosotros de manera explícita.



```

/* SELECT
  NEW es.ubu.lsi.model.asociacion.TipoIncidenciaRanking ( t.descripcion,
  COUNT(i))
FROM
  TipoIncidencia t
LEFT JOIN
  t.incidencias i
GROUP BY
  t.descripcion
ORDER BY
  count(i) */ select
  tipoincide0_.descripcion as col_0_0_,
  count(incidencia1_.fecha) as col_1_0_
from
  HR.TipoIncidencia tipoincide0_
left outer join
  HR.Incidencia incidencia1_
    on tipoincide0_.id=incidencia1_.idtipo
group by
  tipoincide0_.descripcion
order by
  count(incidencia1_.fecha)

```

Como dato curioso, observamos que el count se realiza mediante la fecha.

## Método 6 – Consultar Asociaciones.

Al utilizar un grafo, se aplica la estrategia de carga JOIN. De esta manera se concentran todos los datos en una sola consulta, siendo más eficiente:

```

/* Asociacion.findAll */ select
  asociacion0_.idasoc as idasoc1_0_0_,
  conductor2_.nif as nif1_2_1_,
  incidencia3_.fecha as fecha1_3_2_,
  incidencia3_.nif as nif2_3_2_,
  tipoincide4_.id as id1_4_3_,
  asociacion0_.direccion as direccion2_0_0_,
  asociacion0_.ciudad as ciudad3_0_0_,
  asociacion0_.CP as cp4_0_0_,
  asociacion0_.nombre as nombre5_0_0_,
  conductor2_.apellido as apellido2_2_1_,
  conductor2_.direccion as direccion3_2_1_,
  conductor2_.ciudad as ciudad4_2_1_,
  conductor2_.CP as cp5_2_1_,
  conductor2_.nombre as nombre6_2_1_,
  conductor2_.puntos as puntos7_2_1_,
  conductore1_.idasoc as idasoc1_1_0_,
  conductore1_.nif as nif2_1_0_,
  incidencia3_.anotacion as anotacion3_3_2_,
  incidencia3_.idtipo as idtipo4_3_2_,
  incidencia3_.nif as nif2_3_1_,
  incidencia3_.fecha as fecha1_3_1_,
  tipoincide4_.descripcion as descripcion2_4_3_,
  tipoincide4_.valor as valor3_4_3_
from
  HR.Asociacion asociacion0_
left outer join
  HR.Asociacion_Conductor conductore1_
    on asociacion0_.idasoc=conductore1_.idasoc
left outer join
  HR.Conductor conductor2_
    on conductore1_.nif=conductor2_.nif
left outer join
  HR.Incidencia incidencia3_
    on conductor2_.nif=incidencia3_.nif
left outer join
  HR.TipoIncidencia tipoincide4_
    on incidencia3_.idtipo=tipoincide4_.id
order by
  asociacion0_.idasoc

```

## Consideraciones

Se puede apreciar que , ciertos métodos resurgen un problema de N+1 pudiéndose solucionarse mediante el cambio del tipo de carga de los datos.



Sin embargo, consideramos que , aunque el numero de operaciones crece considerablemente, depende de las circunstancias a las que se suscriba la aplicación no existiendo una solución “mejor” que la otra.

En nuestro caso, al no indicar nada en enunciado , no hemos realizado modificaciones en el tipo de carga de los datos.

## Bibliografía

**\*\*Cabe destacar que la realización del trabajo está fechada a 30/05/21 por lo que la validez de los datos y links aquí mostrados se suscriben a este hecho.**

**Todo el contenido aquí expuesto, o se ha formalizado mediante la información extraída de las fuentes expuestas a continuación, o ha sido mediante reflexiones propias sobre los resultados mostrados por pantalla y conocimientos propios. Todas las imagen mostradas , o han sido capturadas por nosotros, o se pueden encontrar en los enlaces de a continuación. \*\***

Apuntes de la asignatura de Aplicación de Bases de Datos.