

Contenido

Salida de las consultas	2
Explicación	5
1. Insertar incidencia correcta	5
Operaciones	5
Conclusión	5
2. Insertar incidencia con tipo erróneo	5
Operaciones	5
Conclusión	5
3. Indulto del conductor	5
Operaciones	5
Conclusión	6
4. Indultar a un conductor que no existe	6
Operaciones	6
Conclusión	6
5. Información completa con grafos de entidades	6
Operaciones	6
Conclusión	6
Ventajas y desventajas	6

Salida de las consultas

```

JPA Details Console X
<terminated> TestClient [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (17 may. 2025 10:30:29)
Iniciando...
Probando el servicio...
Framework y servicio iniciado...
Insertar incidencia correcta
Hibernate:
select
  conductor0_.nif as nif1_0_0_,
  conductor0_.apellido as apellido2_0_0_,
  conductor0_.ciudad as ciudad3_0_0_,
  conductor0_.cp as cp4_0_0_,
  conductor0_.direccion as direccion5_0_0_,
  conductor0_.nombre as nombre6_0_0_,
  conductor0_.puntos as puntos7_0_0_,
  conductor0_.IDAUTO as idauto8_0_0_,
  vehiculo1_.idauto as idauto1_3_1_,
  vehiculo1_.ciudad as ciudad2_3_1_,
  vehiculo1_.cp as cp3_3_1_,
  vehiculo1_.direccion as direccion4_3_1_,
  vehiculo1_.nombre as nombre5_3_1_
from
  HR.Conductor conductor0_
left outer join
  HR.Vehiculo vehiculo1_
    on conductor0_.IDAUTO=vehiculo1_.idauto
where
  conductor0_.nif=?
Hibernate:
select
  tipoincideo_.id as id1_2_0_,
  tipoincideo_.descripcion as descripcion2_2_0_,
  tipoincideo_.valor as valor3_2_0_
from
  HR.TipoIncidencia tipoincideo_
where
  tipoincideo_.id=?

Hibernate:
select
  incidencia0_.NIF as nif2_1_0_,
  incidencia0_.fecha as fecha1_1_0_,
  incidencia0_.fecha as fecha1_1_1_,
  incidencia0_.NIF as nif2_1_1_,
  incidencia0_.anotacion as anotacion3_1_1_,
  incidencia0_.IDTIPO as idtipo4_1_1_,
  tipoincidel_.id as id1_2_2_,
  tipoincidel_.descripcion as descripcion2_2_2_,
  tipoincidel_.valor as valor3_2_2_
from
  HR.Incidencia incidencia0_
left outer join
  HR.TipoIncidencia tipoincidel_
    on incidencia0_.IDTIPO=tipoincidel_.id
where
  incidencia0_.NIF=?
Hibernate:
/* insert es.ubu.lsi.model.multas.Incidencia
*/ insert
into
  HR.Incidencia
  (IDTIPO, fecha, NIF, anotacion)
values
  (?, ?, ?, ?)
Hibernate:
/* update
es.ubu.lsi.model.multas.Conductor */ update
  HR.Conductor
set
  apellido=?,
  ciudad=?,
  cp=?,
  direccion=?,
  nombre=?,
  puntos=?,
  IDAUTO=?
where
  nif=?
OK incidencia bien insertada
OK actualiza bien los puntos del conductor
Insertar incidencia con tipo erróneo
Hibernate:
select
  conductor0_.nif as nif1_0_0_,
  conductor0_.apellido as apellido2_0_0_,
  conductor0_.ciudad as ciudad3_0_0_,
  conductor0_.cp as cp4_0_0_,
  conductor0_.direccion as direccion5_0_0_,
  conductor0_.nombre as nombre6_0_0_,
  conductor0_.puntos as puntos7_0_0_,
  conductor0_.IDAUTO as idauto8_0_0_,
  vehiculo1_.idauto as idauto1_3_1_,
  vehiculo1_.ciudad as ciudad2_3_1_,
  vehiculo1_.cp as cp3_3_1_,
  vehiculo1_.direccion as direccion4_3_1_,
  vehiculo1_.nombre as nombre5_3_1_
from
  HR.Conductor conductor0_
left outer join
  HR.Vehiculo vehiculo1_
    on conductor0_.IDAUTO=vehiculo1_.idauto
where
  conductor0_.nif=?
Hibernate:
select
  tipoincideo_.id as id1_2_0_,
  tipoincideo_.descripcion as descripcion2_2_0_,
  tipoincideo_.valor as valor3_2_0_
from
  HR.TipoIncidencia tipoincideo_

```

```

from
    HR.TipoIncidencia tipoincide0_
where
    tipoincide0_.id=?
    OK detecta correctamente que NO existe ese tipo de incidencia
Indulto del conductor...
Hibernate:
select
    conductor0_.nif as nif1_0_0_,
    conductor0_.apellido as apellido2_0_0_,
    conductor0_.ciudad as ciudad3_0_0_,
    conductor0_.cp as cp4_0_0_,
    conductor0_.direccion as direccion5_0_0_,
    conductor0_.nombre as nombre6_0_0_,
    conductor0_.puntos as puntos7_0_0_,
    conductor0_.IDAUTO as idauto8_0_0_,
    vehiculo1_.idauto as idauto1_3_1_,
    vehiculo1_.ciudad as ciudad2_3_1_,
    vehiculo1_.cp as cp3_3_1_,
    vehiculo1_.direccion as direccion4_3_1_,
    vehiculo1_.nombre as nombre5_3_1_
from
    HR.Conductor conductor0_
left outer join
    HR.Vehiculo vehiculo1_
        on conductor0_.IDAUTO=vehiculo1_.idauto
where
    conductor0_.nif=?
Hibernate:
/* Incidencia.deleteAllWithNIF */ delete
from
    HR.Incidencia
where
    NIF=?
Hibernate:
select
    incidencia0_.NIF as nif2_1_0_,

```

```

Hibernate:
select
    incidencia0_.NIF as nif2_1_0_,
    incidencia0_.fecha as fecha1_1_0_,
    incidencia0_.fecha as fecha1_1_1_,
    incidencia0_.NIF as nif2_1_1_,
    incidencia0_.anotacion as anotacion3_1_1_,
    incidencia0_.IDTIPO as idtipo4_1_1_,
    tipoincide1_.id as id1_2_2_,
    tipoincide1_.descripcion as descripcion2_2_2_,
    tipoincide1_.valor as valor3_2_2_
from
    HR.Incidencia incidencia0_
left outer join
    HR.TipoIncidencia tipoincide1_
        on incidencia0_.IDTIPO=tipoincide1_.id
where
    incidencia0_.NIF=?
Hibernate:
/* update
es.ubu.lsi.model.multas.Conductor */ update
    HR.Conductor
set
    apellido=?,
    ciudad=?,
    cp=?,
    direccion=?,
    nombre=?,
    puntos=?,
    IDAUTO=?
where
    nif=?
    OK todas las incidencias borradas del conductor indultado
    OK puntos bien iniciados con indulto
    OK el número de incidencias de otros conductores es correcto
Indultar a un conductor que no existe
Hibernate:
select

```

<terminated> TestClient [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (17 may. 2025 10:30:29)

indultar a un conductor que no existe

```

Hibernate:
select
    conductor0_.nif as nif1_0_0_,
    conductor0_.apellido as apellido2_0_0_,
    conductor0_.ciudad as ciudad3_0_0_,
    conductor0_.cp as cp4_0_0_,
    conductor0_.direccion as direccion5_0_0_,
    conductor0_.nombre as nombre6_0_0_,
    conductor0_.puntos as puntos7_0_0_,
    conductor0_.IDAUTO as idauto8_0_0_,
    vehiculo1_.idauto as idauto1_3_1_,
    vehiculo1_.ciudad as ciudad2_3_1_,
    vehiculo1_.cp as cp3_3_1_,
    vehiculo1_.direccion as direccion4_3_1_,
    vehiculo1_.nombre as nombre5_3_1_
from
    HR.Conductor conductor0_
left outer join
    HR.Vehiculo vehiculo1_
        on conductor0_.IDAUTO=vehiculo1_.idauto
where
    conductor0_.nif=?
    OK detecta correctamente que NO existe ese conductor
Información completa con grafos de entidades...
Hibernate:
/* Vehiculo.findAll */ select
    vehiculo0_.idauto as idauto1_3_0_,
    conductore1_.nif as nif1_0_1_,
    incidencia2_.fecha as fecha1_1_2_,
    incidencia2_.NIF as nif2_1_2_,
    tipoincide3_.id as id1_2_3_,
    vehiculo0_.ciudad as ciudad2_3_0_,
    vehiculo0_.cp as cp3_3_0_,
    vehiculo0_.direccion as direccion4_3_0_,
    vehiculo0_.nombre as nombre5_3_0_,
    conductore1_.apellido as apellido2_0_1_,
    conductore1_.ciudad as ciudad3_0_1_,

```

```
JPA Details Console
<terminated> TestClient [Java Application] /usr/lib/jvm/java-11-openjdk-amd64/bin/java (17 may. 2025 10:30:29)
vehiculo0.nombre as nombres_3_0_,
conductor01.apellido as apellido2_0_1_,
conductor01.ciudad as ciudad3_0_1_,
conductor01.cp as cp4_0_1_,
conductor01.direccion as direccion5_0_1_,
conductor01.nombre as nombre6_0_1_,
conductor01.puntos as puntos7_0_1_,
conductor01.IDAUTO as idauto8_0_1_,
conductor01.IDAUTO as idauto8_0_0_,
conductor01.nif as nif1_0_0_,
incidencia2.anotacion as anotacion3_1_2_,
incidencia2.IDTIPO as idtipo4_1_2_,
incidencia2.NIF as nif2_1_1_,
incidencia2.fecha as fecha1_1_1_,
tipoincide3.descripcion as descripcion2_2_3_,
tipoincide3.valor as valor3_2_3_
from
HR.Vehiculo vehiculo0_
left outer join
HR.Conductor conductor01_
on vehiculo0_.idauto=conductor01_.IDAUTO
left outer join
HR.Incidencia incidencia2_
on conductor01_.nif=incidencia2_.NIF
left outer join
HR.TipoIncidencia tipoincide3_
on incidencia2_.IDTIPO=tipoincide3_.id
es.ubu.lsi.model.multas.Vehiculo@36df4c26
Conductor [nif=10000000A, nombre=Juana, apellido=Manzanal, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@5a4c98d7, puntos=3]
es.ubu.lsi.model.multas.Incidencia@38732372
es.ubu.lsi.model.multas.Incidencia@2fee69a1
Conductor [nif=10000000C, nombre=Jimena, apellido=Plaza, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@668a32a4, puntos=12]
Conductor [nif=10000000B, nombre=Javier, apellido=Calle, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@2026fbff, puntos=6]
es.ubu.lsi.model.multas.Incidencia@6a4af081
es.ubu.lsi.model.multas.Vehiculo@36b53f08
Conductor [nif=20000000C, nombre=Pablo, apellido=Torquemada, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@baa227e, puntos=3]
es.ubu.lsi.model.multas.Incidencia@590013c7
es.ubu.lsi.model.multas.Incidencia@64f32e9e
es.ubu.lsi.model.multas.Incidencia@64f32e9e
Conductor [nif=20000000A, nombre=Paloma, apellido=Del Barrio, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@351e89fc, puntos=12]
Conductor [nif=20000000B, nombre=Pedro, apellido=Medina, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@2b682e9, puntos=12]
es.ubu.lsi.model.multas.Vehiculo@7b211077
Conductor [nif=30000000A, nombre=Raquel, apellido=Del Barrio, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@15586843, puntos=9]
es.ubu.lsi.model.multas.Incidencia@62b0bf85
Conductor [nif=30000000C, nombre=Roberto, apellido=Manzanita, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@4fbb1144, puntos=0]
es.ubu.lsi.model.multas.Incidencia@734fbae3
Conductor [nif=30000000B, nombre=Rosa, apellido=Manzanedo, direccionPostal=es.ubu.lsi.model.multas.DireccionPostal@e2b3026, puntos=6]
es.ubu.lsi.model.multas.Incidencia@6c9bf3b5
OK Sin excepciones en la consulta completa y acceso posterior
FIN.....
```

Explicación

1. Insertar incidencia correcta

Operaciones

- SELECT HR.Conductor
- SELECT HR.TipoIncidencia
- SELECT HR.Incidencia
- INSERT INTO HR.Incidencia
- UPDATE HR.Conductor

Resultado de las operaciones: 3 selects, 1 insert y un 1 update, lo que nos da un total de 5 operaciones

Conclusión

Las operaciones de select son para cargar las entidades en el contexto de persistencia de hibernate la operación de insert crea una nueva incidencia y la operación de update es para actualizar los datos del conductor teniendo en cuenta la incidencia

2. Insertar incidencia con tipo erróneo

Operaciones

- SELECT HR.Conductor
- SELECT HR. TipoIncidencia

Resultado de las operaciones: 2 selects lo que nos da un total de 2 operaciones

Conclusión

Hibernate carga la entidades de conductor y de tipo de incidencia y al no encontrar tipo de incidencia previene la inserción, lo sabemos porque no hay ningún insert

3. Indulto del conductor

Operaciones

- SELECT HR.Conductor
- DELETE FROM HR.Incidencia
- SELECT HR.Incidencia
- UPDATE HR.Conductor

Resultado de las operaciones: 2 selects, 1 delete y un 1 update, lo que nos da un total de 4 operaciones

Conclusión

El select de conductor carga la entidad en hibernate, el delete ejecuta la eliminación de la incidencia y el siguiente select es para obtener los datos de las incidencias para mas tarde actualizar al conducto con los nuevos datos.

4. Indultar a un conductor que no existe

Operaciones

- SELECT HR.Conductor

Resultado de las operaciones: 1 selects lo que nos da un total de 1 operación

Conclusión

Hibernate intenta cargar la entidad conductor como no existe, no se realizan más operaciones

5. Información completa con grafos de entidades

Operaciones

- /* Vehiculo.findAll */SELECT vehiculo0_.idauto ..., conductore1_.nif ..., incidencia2_.fecha ..., tipoincidence3_.id

Resultado de las operaciones: 1 selects lo que nos da un total de 1 operación

Conclusión

Hibernate selecciona todos los datos de todas las entidades esta seleccionando los datos a partir de la entidad de vehículo utilizando left outer join de las tablas conductor, incidencia y tipoIncidencia

Ventajas y desventajas

Las ventajas claras son que nos permite trabajar directamente con objetos lo que nos facilita mucho su implementación en el código, a su mismo tiempo esto añade una curva de aprendizaje más compleja puesto que ya no está fácil como en otras prácticas que tirábamos las consultas directamente. Una desventaja que a mi se me ocurre es que puede llegar a suceder que las consultas que se estén generando con esta abstracción sean menos optimas que si las programáramos a mano. Otras ventajas que

nos otorga hibernate es la prevención de SQL, el cacheo de las entidades y que el ahora es el que se encarga de gestionar el ciclo de vida de las entidades y relaciones.

En mi forma de verlo me parece más fácil de utilizar JDBC puro como lo usamos en la práctica de donación de sangre me parece que es mucho más simple y fácil de entender pero entiendo que cuando tienes una aplicación muy compleja puede ser muy útil implementar algún tipo de abstracción.