

ciclo: [DAM]

MÓDULO DE [BASES DE DATOS]

[Tarea Nº 05]

Alumno:

[Juan Carlos Filter Martín]

[15456141A]

**Contenido**

[1. Documentos que se adjuntan a este informe. 3](#__RefHeading___Toc88_3906032408)

[2. Diagrama Entidad-Relación del esquema HR 3](#__RefHeading___Toc92_3906032408_Copia_1)

[3. (RA04\_b) Se han insertado, borrado y actualizado datos en las tablas 4](#__RefHeading___Toc92_3906032408)

[1. Inserción: 4](#__RefHeading___Toc94_3906032408)

[a) Inserte un empleado con el nombre (first\_name) y los apellidos (last\_name) del autor de la tarea y con salario de 30000 en el departamento 100 y el resto de los datos a tu elección. 4](#__RefHeading___Toc252_3564960023)

[b) Inserte un empleado llamado Juan Márquez con salario de 35000 en el departamento 100 y el resto de los datos a tu elección. 4](#__RefHeading___Toc254_3564960023)

[c) Consulta los datos de empleados para ver que se han introducido correctamente. 4](#__RefHeading___Toc256_3564960023)

[d) Crea un punto de salvado llamado INSERTADO. 5](#__RefHeading___Toc258_3564960023)

[2. Modificación: 5](#__RefHeading___Toc96_3906032408)

[a) Modifique el salario de los empleados que sea superior a 20000 disminuyéndolo en 1000. 5](#__RefHeading___Toc260_3564960023)

[b) Modifique el id de departamento a 110 para los empleados cuyo apellido acabe en "ez". 6](#__RefHeading___Toc262_3564960023)

[c) Consulta los datos de empleados para ver que se han modificado correctamente. 6](#__RefHeading___Toc264_3564960023)

[d) Crea un punto de salvado llamado MODIFICADO. 7](#__RefHeading___Toc266_3564960023)

[3. Eliminación: 7](#__RefHeading___Toc98_3906032408)

[a) Elimine el empleado con nombre y apellidos iguales a los tuyos. 7](#__RefHeading___Toc268_3564960023)

[b) Elimine los empleado con salario mayor o igual que 30000 y que sean del departamento 110. 7](#__RefHeading___Toc270_3564960023)

[c) Si en alguna de las eliminaciones tenemos un problema de integridad referencial, debemos resolverlo y documentarlo. 8](#__RefHeading___Toc272_3564960023)

[d) Consulta los datos de empleados para ver que se han eliminado correctamente. 10](#__RefHeading___Toc274_3564960023)

[e) Crea un punto de salvado llamado ELIMINADO 10](#__RefHeading___Toc276_3564960023)

[4. (RA04\_c) Se ha incluido en una tabla la información resultante de la ejecución de una consulta. 11](#__RefHeading___Toc100_3906032408)

[a) Modifica el salario de todos los empleados por el salario máximo de su categoría profesional. Columna 'max\_salary' de tabla JOBS. 11](#__RefHeading___Toc278_3564960023)

[b) Consulta los datos de empleados para ver que se han modificado correctamente 11](#__RefHeading___Toc280_3564960023)

[c) Crea un punto de salvado llamado MODIFICA\_TODO. 12](#__RefHeading___Toc282_3564960023)

[5. (RA04\_f) Se han anulado parcial o totalmente los cambios producidos por una transacción. 13](#__RefHeading___Toc102_3906032408)

[a) Vuelve al punto de salvado MODIFICADO y comprueba que no se han eliminado los empleados. ¿Qué ocurre? 13](#__RefHeading___Toc284_3564960023)

[b) Vuelve al punto de salvado ELIMINADO ¿Qué ocurre? 14](#__RefHeading___Toc286_3564960023)

[c) Vuelve al inicio de la transacción. Comprueba y explica que ha pasado con los datos introducidos, modificados y eliminados. 14](#__RefHeading___Toc288_3564960023)

[d) Finaliza y confirma la transacción. 15](#__RefHeading___Toc290_3564960023)

[6. (RA04\_d) Se han diseñado guiones de sentencias para llevar a cabo tareas complejas. 16](#__RefHeading___Toc104_3906032408)

[Crear un script SQL (un fichero con extensión .sql) 16](#__RefHeading___Toc292_3564960023)

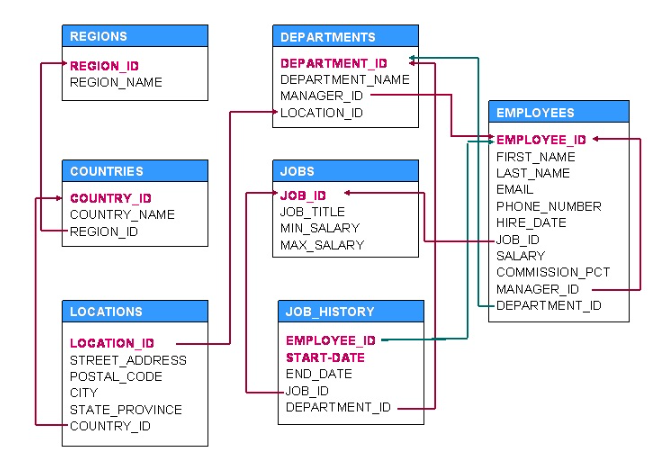
# Documentos que se adjuntan a este informe.

A continuación se detallan los documentos que componen la presente entrega de la tarea:

1. Informe de elaboración de la tarea.
2. Archivo ScripSQL.sql

# Diagrama Entidad-Relación del esquema HR

Este es el diagrama E-R del esquema HR para ver las uniones entre tablas de esta tarea



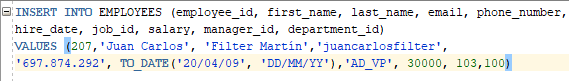
# (RA04\_b) Se han insertado, borrado y actualizado datos en las tablas

## Inserción:

### **Inserte un empleado con el nombre (first\_name) y los apellidos (last\_name) del autor de la tarea y con salario de 30000 en el departamento 100 y el resto de los datos a tu elección.**

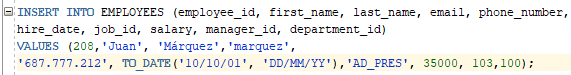
Mediante INSERT INTO (para insertar datos) seguido de la tabla EMPLOYES le indicamos los campos/columnas que vamos a introducir, seguido de VALUES y los datos.

He ignorado columnas como por ejemplo COMMISSION\_PCT entonces su valor será reflejado como null.



### **Inserte un empleado llamado Juan Márquez con salario de 35000 en el departamento 100 y el resto de los datos a tu elección.**

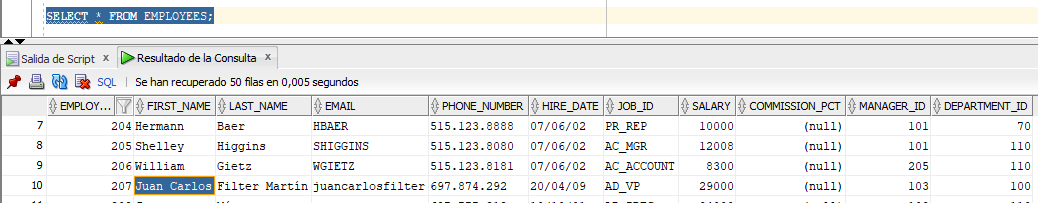
Igual que en el punto anterior tenemos que introducir nuevos datos entonces se hace mediante un INSERT INTO pero en este caso con el nombre de Juan, apellido Márquez, un salario de 35000 y el departamento 100



### **Consulta los datos de empleados para ver que se han introducido correctamente.**

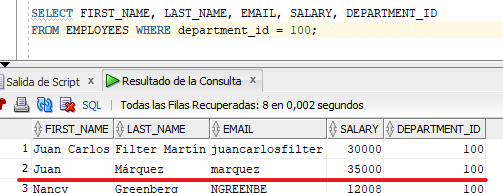
* + Comprobamos los datos insertados en la tabla EMPLOYEES:

SELECT \* FROM EMPLOYEES



SELECT FIRST\_NAME, LAST\_NAME, EMAIL, SALARY, DEPARTMENT\_ID

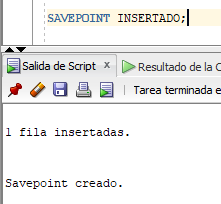
FROM EMPLOYEES WHERE department\_id = 100



### **Crea un punto de salvado llamado INSERTADO.**

La sentencia SAVEPOINT marca un punto de grabación en la transacción actual.

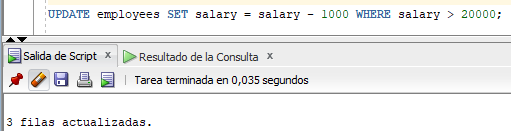
Entonces mediante SAVEPOINT INSERTADO estamos creado un punto de grabación con el nombre INSERTADO.

**

## Modificación:

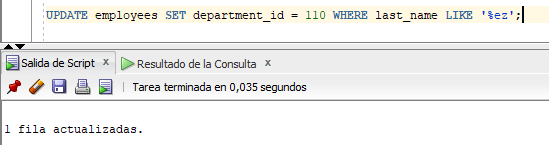
### **Modifique el salario de los empleados que sea superior a 20000 disminuyéndolo en 1000.**

Podemos modificar con la sentencia UPDATE de la tabla EMPLOYEES le decimos que salary sea igual a salary -1000 donde salary sea mayor a 20000



### **Modifique el id de departamento a 110 para los empleados cuyo apellido acabe en "ez".**

Con la sentencia UPDATE en la tabla EMPLOYEES le indicamos que modifique el id del departamento a 110 siempre que se cumpla (con WHERE) que el apellido acabe en ‘%ez’

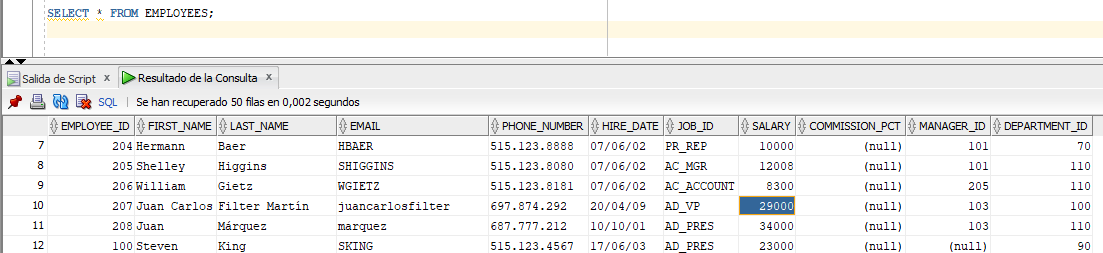


### **Consulta los datos de empleados para ver que se han modificado correctamente.**

SELECT \* FROM EMPLOYEES

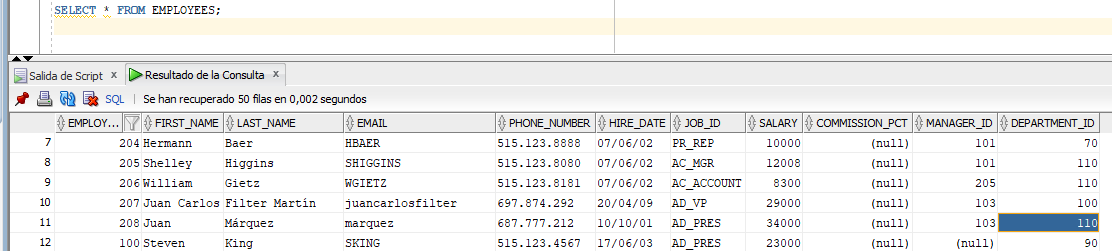
Primera modificación:

Podemos comprobar que tanto mis datos que tenían un salario de 30000 como el de Juan Márquez que eran de 35000 ahora se le han restado 1000.



Segunda modificación:

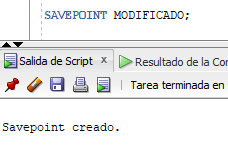
Podemos comprobar que Juan Marquéz estaba asignado en el department\_id 100 y tras el update que hicimos ahora está asignado en el department\_id 110



### **Crea un punto de salvado llamado MODIFICADO.**

Como en el primer apartado de insertado realizamos un punto de salvado con la sentencia SAVEPOINT

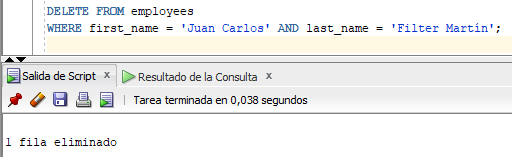
SAVEPOINT MODIFICADO estamos creado un punto de grabación con el nombre MODIFICADO.



## Eliminación:

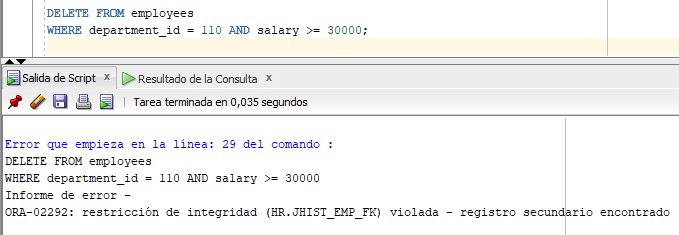
### **Elimine el empleado con nombre y apellidos iguales a los tuyos.**

Mediante la sentencia DELETE indicamos la tabla EMPLOYEES y con un WHERE filtramos que el nombre sea ‘Juan Carlos’ y el apellido ‘Filter Martín’



### **Elimine los empleado con salario mayor o igual que 30000 y que sean del departamento 110.**

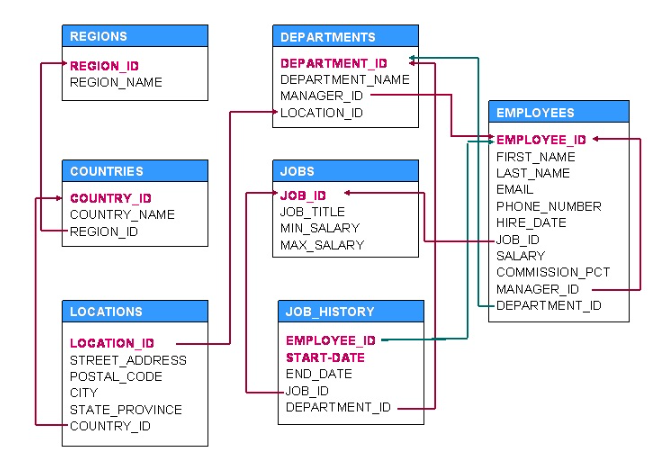
Al indicarle que elimine los empleados que se encuentren en el department\_id 100 y tengan un salary de 30000 o mayor aparece el siguiente error de integridad



**Error de integridad resuelto en el siguiente punto**

### **Si en alguna de las eliminaciones tenemos un problema de integridad referencial, debemos resolverlo y documentarlo.**

Lo primero es ver el diagrama E-R



Viendo este diagrama lo primero que vamos a hacer es lo siguiente:

* + - Identificar la clave externa:



El propio error nos dá una pista diciendo que JHIST\_EMP\_FK. ¿y que significa este constraint?

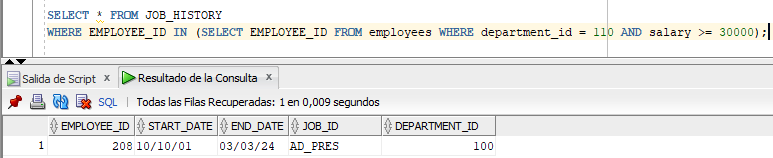
Pues si vamos al diagrama y con las abreviaciones dada podemos deducir que Job\_history tiene fk a employees.

* + - Realizar una consulta:

Con SELECT que muestre toda la tabla JOB\_HISTORY que es donde se encuentra la clave externa employee\_id cumpla con la siguiente subconsulta que será la siguiente:

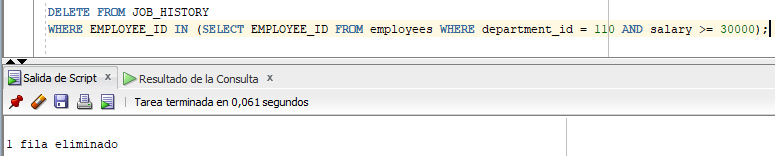
En la tabla empleado donde department\_id sea 110 y el salary sea mayor o igual a 30000

Y conseguimos el siguiente resultado:

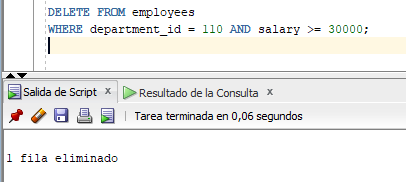


* + - Realizar el DELETE:

Si ahora usamos el DELETE borraremos los datos que se mostraron en la consulta.

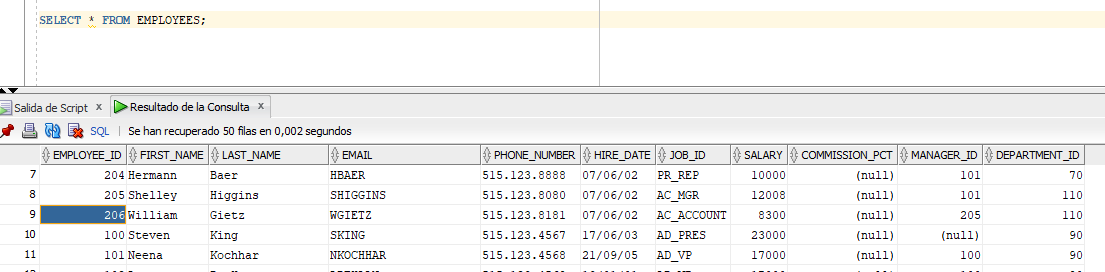


Por último si realizamos la sentencia DELETE del punto anterior si que podremos borrarlo.

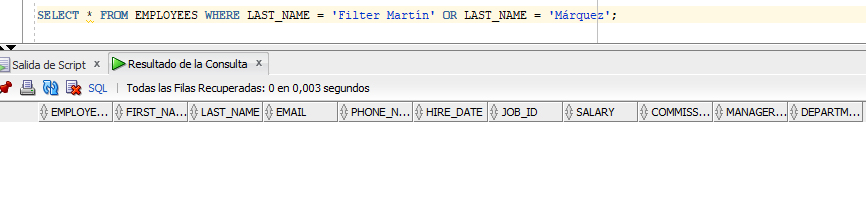


### **Consulta los datos de empleados para ver que se han eliminado correctamente.**

SELECT \* FROM EMPLOYEES

Ambos datos ya no están en la tabla empleado.

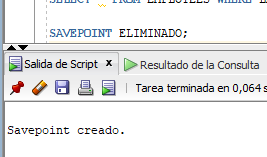
Podemos comprobar mediante el WHERE filtrando por apellido ‘Filter Martín’ como ‘Márquez’ no aparece ningún registro.



### **Crea un punto de salvado llamado ELIMINADO**

Realizamos un punto de salvado con la sentencia SAVEPOINT

SAVEPOINT ELIMINADO estamos creado un punto de grabación con el nombre ELIMINADO.

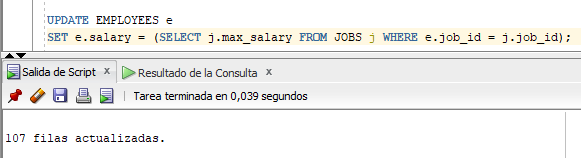
****

# (RA04\_c) Se ha incluido en una tabla la información resultante de la ejecución de una consulta.

### **Modifica el salario de todos los empleados por el salario máximo de su categoría profesional. Columna 'max\_salary' de tabla JOBS.**

Mediante un UPDATE a la tabla employees con alias ”e” le indicamos que modifique el salary de empleado que sea igual a la siguiente consulta:

consulta sobre el salario max de jobs con sus alias “j” donde el job\_id de employees y jobs sean iguales.

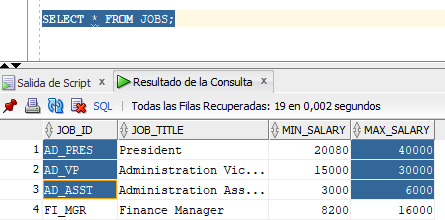


### **Consulta los datos de empleados para ver que se han modificado correctamente**

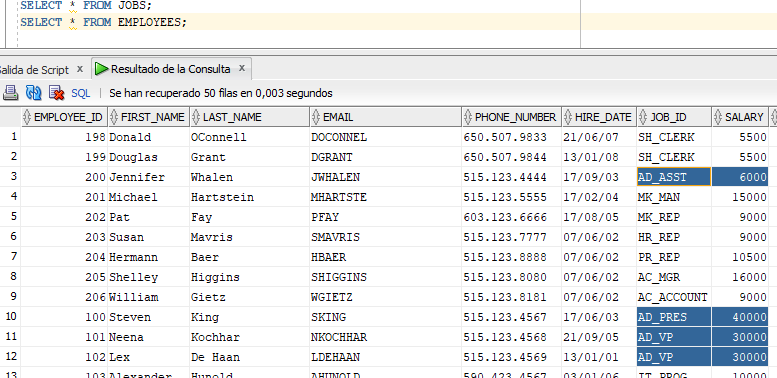
* Resultado:

Podemos comprobarlo mirando distintas tablas (jobs y empleado)

**JOBS**



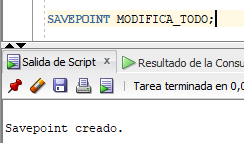
**EMPLOYEES**



### **Crea un punto de salvado llamado MODIFICA\_TODO.**

Realizamos un punto de salvado con la sentencia SAVEPOINT

SAVEPOINT MODIFICA\_TODO estamos creado un punto de grabación con el nombre MODIFICA\_TODO.

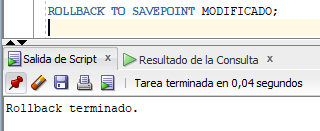


# (RA04\_f) Se han anulado parcial o totalmente los cambios producidos por una transacción.

### **Vuelve al punto de salvado MODIFICADO y comprueba que no se han eliminado los empleados. ¿Qué ocurre?**

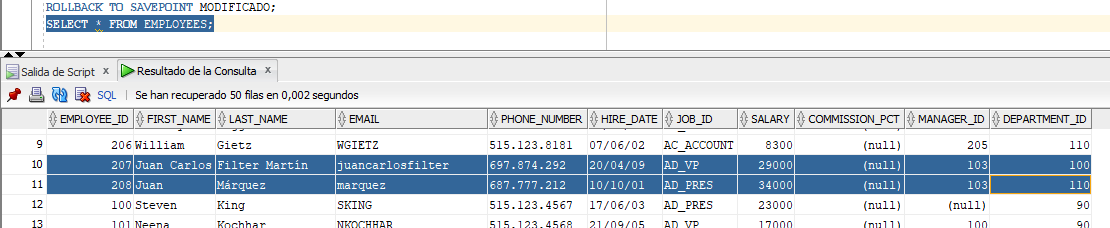
Para volver a un punto de salvado hay que introducir la sentencia:

ROLLBACK TO SAVEPOINT seguido del nombre en este caso MODIFICADO.



Una vez realizado el ROLLBACK al punto de salvado MODIFICADO podemos ver que la ultima modificación es hasta que se hizo el savepoint modificado, lo que significa que el punto eliminado no se han realizado.

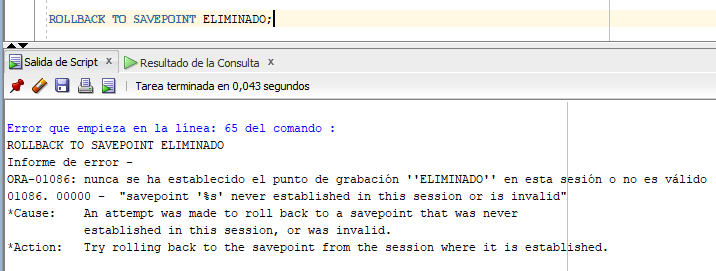
Si hacemos un SELECT \* FROM EMPLOYEES podemos comprobar que por ejemplo mi nombre y el de Juan Márquez si aparecen (Estos fueron borrado en el apartado eliminado)



### **Vuelve al punto de salvado ELIMINADO ¿Qué ocurre?**

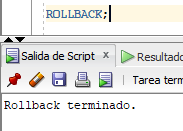
En este caso al hacer el ROLLBACK TO SAVEPOINT ELIMINADO nos sale el siguiente error y esto es porque actualmente estamos en el savepoint MODIFICADO lo que quiere decir que el apartado eliminado no está creado.

Para hacer un ROLLBACK TO SAVEPOINT ELIMINADO habría que introducir nuevamente las sentencias del punto eliminado y su savepoint ELIMINADO.



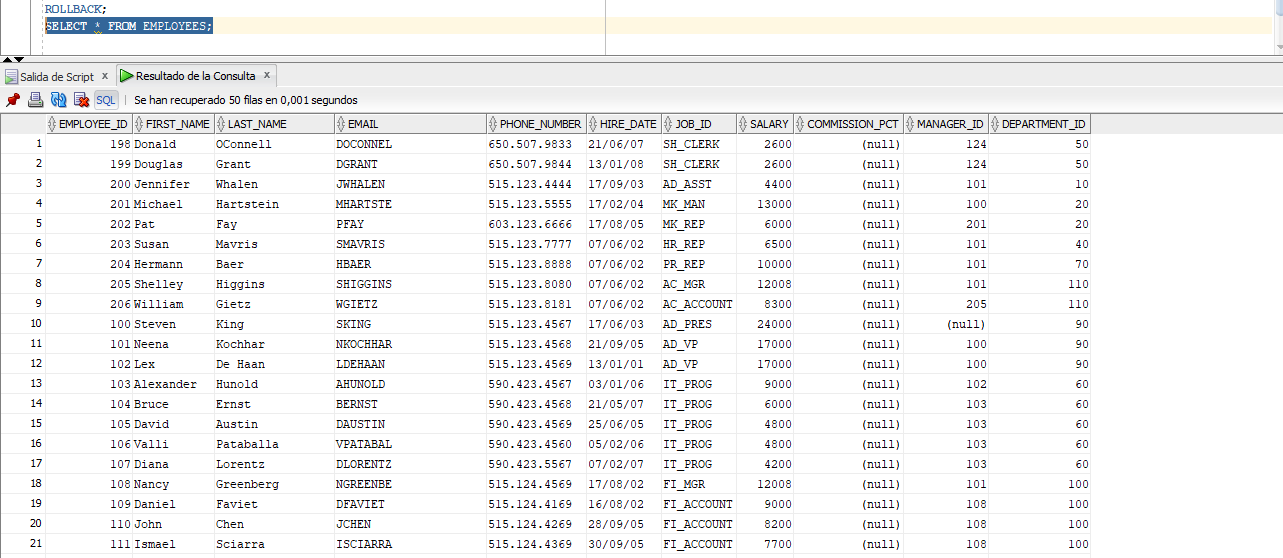
### **Vuelve al inicio de la transacción. Comprueba y explica que ha pasado con los datos introducidos, modificados y eliminados.**

Para volver al inicio de la transacción solamente tendríamos que poner ROLLBACK sin indicar el savepoint.

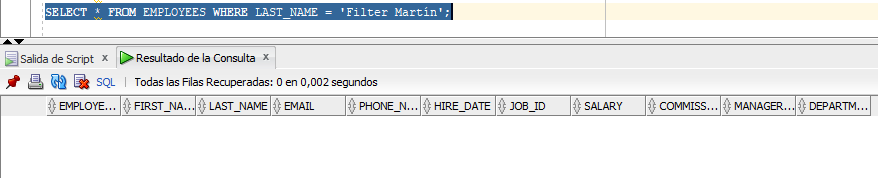


Lo que ha ocurrido al realizar un ROLLBACK es que todos los datos introducidos han sido eliminado, tantos los insertados, modificados como los eliminados.

* Se puede comprobar mediante un SELECT\* FROM EMPLOYEES



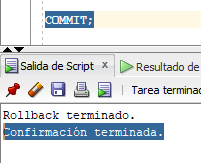
* O bien podemos hacer una consulta filtrando por ejemplo last\_name = ‘Filter Martín’



Podemos comprobar que no hay datos.

### **Finaliza y confirma la transacción.**

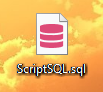
Para finalizar y confirmar la transacción tenemos la sentencian COMMIT



Al hacer un COMMIT se guardan permanentemente los cambios realizados.

# (RA04\_d) Se han diseñado guiones de sentencias para llevar a cabo tareas complejas.

### **Crear un script SQL (un fichero con extensión .sql)**



Script SQL creado con el orden exactamente igual al informe.

