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

2. Diagrama Entidad-Relación del esquema HR.....3

3. (RA04_b) Se han insertado, borrado y actualizado datos en las tablas.....4

1. Inserción:.....4

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

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

c) Consulta los datos de empleados para ver que se han introducido correctamente.....4

d) Crea un punto de salvado llamado INSERTADO.....5

2. Modificación:.....5

a) Modifique el salario de los empleados que sea superior a 20000 disminuyéndolo en 1000...5

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

c) Consulta los datos de empleados para ver que se han modificado correctamente.....6

d) Crea un punto de salvado llamado MODIFICADO.....7

3. Eliminación:.....7

a) Elimine el empleado con nombre y apellidos iguales a los tuyos.....7

b) Elimine los empleado con salario mayor o igual que 30000 y que sean del departamento 110.....7

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

d) Consulta los datos de empleados para ver que se han eliminado correctamente.....10

e) Crea un punto de salvado llamado ELIMINADO.....10

4. (RA04_c) Se ha incluido en una tabla la información resultante de la ejecución de una consulta. 11

a) Modifica el salario de todos los empleados por el salario máximo de su categoría profesional. Columna 'max_salary' de tabla JOBS.....11

b) Consulta los datos de empleados para ver que se han modificado correctamente.....11

c) Crea un punto de salvado llamado MODIFICA_TODO.....12

5. (RA04_f) Se han anulado parcial o totalmente los cambios producidos por una transacción....13

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

b) Vuelve al punto de salvado ELIMINADO ¿Qué ocurre?.....14

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

d) Finaliza y confirma la transacción.....15

6. (RA04_d) Se han diseñado guiones de sentencias para llevar a cabo tareas complejas.....16

Crear un script SQL (un fichero con extensión .sql).....16

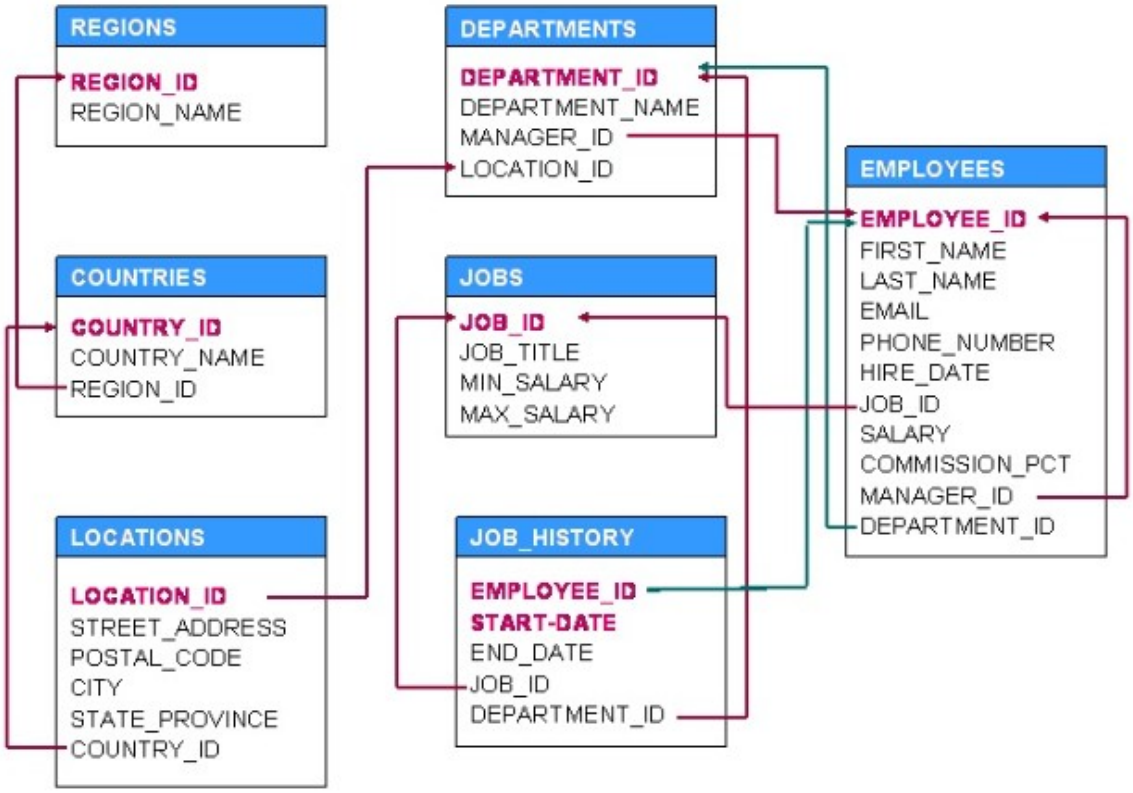
1. Documentos que se adjuntan a este informe.

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

2. Diagrama Entidad-Relación del esquema HR

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



3. (RA04_b) Se han insertado, borrado y actualizado datos en las tablas

1. *Inserción:*

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.

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.

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, manager_id, department_id)
VALUES (207,'Juan Carlos', 'Filter Martín','juancarlosfilter',
'697.874.292', TO_DATE('20/04/09', 'DD/MM/YY'),'AD_VP', 30000, 103,100)
```

b) 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

```
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, manager_id, department_id)
VALUES (208,'Juan', 'Márquez','marquez',
'687.777.212', TO_DATE('10/10/01', 'DD/MM/YY'),'AD_PRES', 35000, 103,100);
```

c) Consulta los datos de empleados para ver que se han introducido correctamente.

- Comprobamos los datos insertados en la tabla EMPLOYEES:

SELECT * FROM EMPLOYEES

```
SELECT * FROM EMPLOYEES;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,005 segundos

	EMPLOY...	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	207	Juan Carlos	Filter Martín	juancarlosfilter	697.874.292	20/04/09	AD_VP	29000	(null)	103	100

SELECT FIRST_NAME, LAST_NAME, EMAIL, SALARY, DEPARTMENT_ID
FROM EMPLOYEES WHERE department_id = 100

```
SELECT FIRST_NAME, LAST_NAME, EMAIL, SALARY, DEPARTMENT_ID
FROM EMPLOYEES WHERE department_id = 100;
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 8 en 0,002 segundos

	FIRST_NAME	LAST_NAME	EMAIL	SALARY	DEPARTMENT_ID
1	Juan Carlos	Filter Martín	juancarlosfilter	30000	100
2	Juan	Márquez	marquez	35000	100
3	Nancy	Greenberg	NGREENBE	12008	100

d) 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.

```
SAVEPOINT INSERTADO;
```

Salida de Script | Resultado de la C

Tarea terminada e

```
1 fila insertadas.

Savepoint creado.
```

2. *Modificación:*

a) 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

```
UPDATE employees SET salary = salary - 1000 WHERE salary > 20000;
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,035 segundos

```
3 filas actualizadas.
```

b) 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'

```
UPDATE employees SET department_id = 110 WHERE last_name LIKE '%ez';
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,035 segundos

```
1 fila actualizadas.
```

c) Consulta los datos de empleados para ver que se han modificado correctamente.

SELECT * FROM EMPLOYEES

<u>Primera modificación:</u>

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.

```
SELECT * FROM EMPLOYEES;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,002 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	207	Juan Carlos	Filter Martín	juancarlosfilter	697.874.292	20/04/09	AD_VP	29000	(null)	103	100
11	208	Juan	Márquez	marquez	687.777.212	10/10/01	AD_PRES	34000	(null)	103	110
12	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	23000	(null)	(null)	90

<u>Segunda modificación:</u>

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

```
SELECT * FROM EMPLOYEES;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,002 segundos

	EMPLOY...	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	207	Juan Carlos	Filter Martín	juancarlosfilter	697.874.292	20/04/09	AD_VP	29000	(null)	103	100
11	208	Juan	Márquez	marquez	687.777.212	10/10/01	AD_PRES	34000	(null)	103	110
12	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	23000	(null)	(null)	90

d) 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.

```
SAVEPOINT MODIFICADO;
```

Salida de Script | Resultado de la Cor
Tarea terminada en

```
Savepoint creado.
```

3. *Eliminación:*

a) 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'

```
DELETE FROM employees
WHERE first_name = 'Juan Carlos' AND last_name = 'Filter Martín';
```

Salida de Script | Resultado de la Consulta
Tarea terminada en 0,038 segundos

```
1 fila eliminado
```

b) 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

```
DELETE FROM employees
WHERE department_id = 110 AND salary >= 30000;
```

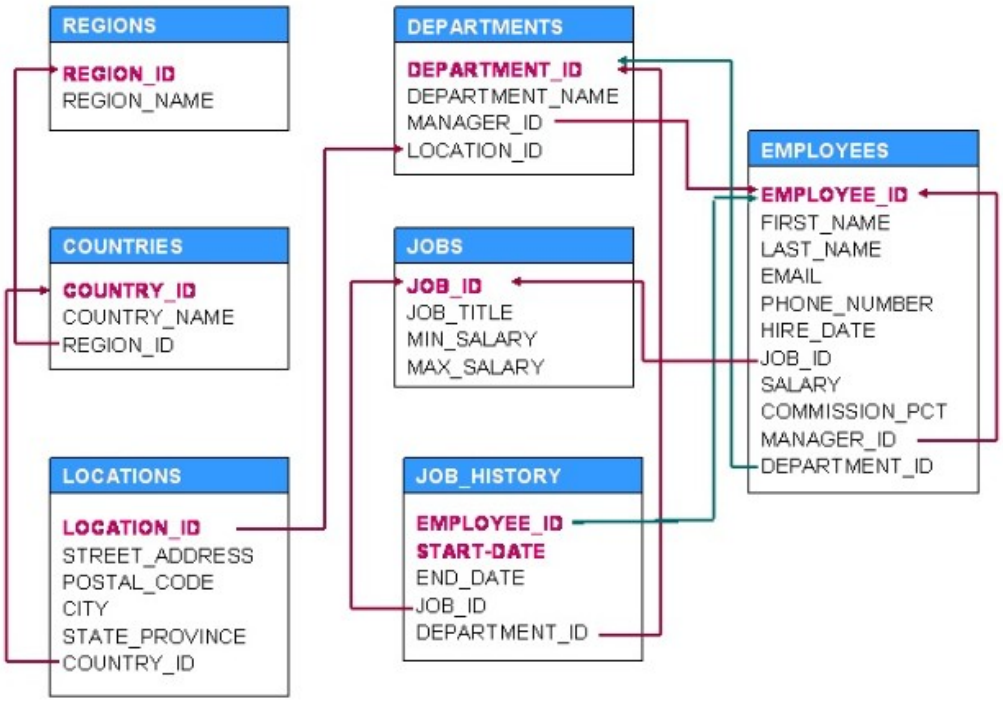
Salida de Script | Resultado de la Consulta
Tarea terminada en 0,035 segundos

```
Error que empieza en la línea: 29 del comando :
DELETE FROM employees
WHERE department_id = 110 AND salary >= 30000
Informe de error -
ORA-02292: restricción de integridad (HR.JHIST_EMP_FK) violada - registro secundario encontrado
```

Error de integridad resuelto en el siguiente punto

c) 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:

- ✗ <u>Identificar la clave externa:</u>

```
Informe de error -
ORA-02292: restricción de integridad (HR.JHIST_EMP_FK) v
```

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:

```
SELECT * FROM JOB_HISTORY
WHERE EMPLOYEE_ID IN (SELECT EMPLOYEE_ID FROM employees WHERE department_id = 110 AND salary >= 30000);
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 1 en 0,009 segundos

	EMPLOYEE_ID	START_DATE	END_DATE	JOB_ID	DEPARTMENT_ID
1	208	10/10/01	03/03/24	AD_PRES	100

✗ Realizar el DELETE:

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

```
DELETE FROM JOB_HISTORY
WHERE EMPLOYEE_ID IN (SELECT EMPLOYEE_ID FROM employees WHERE department_id = 110 AND salary >= 30000);
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,061 segundos

```
1 fila eliminado
```

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

```
DELETE FROM employees
WHERE department_id = 110 AND salary >= 30000;
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,06 segundos

```
1 fila eliminado
```

d) Consulta los datos de empleados para ver que se han eliminado correctamente.

SELECT * FROM EMPLOYEES

Ambos datos ya no están en la tabla empleado.

```
SELECT * FROM EMPLOYEES;
```

Salida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,002 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	23000	(null)	(null)	90
11	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	90

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

```
SELECT * FROM EMPLOYEES WHERE LAST_NAME = 'Filter Martín' OR LAST_NAME = 'Márquez';
```

Salida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 0 en 0,003 segundos

EMPLOYE...	FIRST_NA...	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY	COMMISS...	MANAGER...	DEPARTM...

e) 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.

```
SAVEPOINT ELIMINADO;
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,064 s

```
Savepoint creado.
```

4. (RA04_c) Se ha incluido en una tabla la información resultante de la ejecución de una consulta.

a) 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.

```
UPDATE EMPLOYEES e
SET e.salary = (SELECT j.max_salary FROM JOBS j WHERE e.job_id = j.job_id);
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,039 segundos

```
107 filas actualizadas.
```

b) Consulta los datos de empleados para ver que se han modificado correctamente

✗ Resultado:

Podemos comprobarlo mirando distintas tablas (jobs y empleado)

JOBS

```
SELECT * FROM JOBS;
```

lida de Script | Resultado de la Consulta

SQL | Todas las Filas Recuperadas: 19 en 0,002 segundos

	JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
1	AD_PRES	President	20080	40000
2	AD_VP	Administration Vic...	15000	30000
3	AD_ASST	Administration Ass...	3000	6000
4	FI_MGR	Finance Manager	8200	16000

EMPLOYEES

```
SELECT * FROM JOBS;
SELECT * FROM EMPLOYEES;
```

alida de Script | Resultado de la Consulta

SQL | Se han recuperado 50 filas en 0,003 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
1	198	Donald	OConnell	DOCONNEL	650.507.9833	21/06/07	SH_CLERK	5500
2	199	Douglas	Grant	DGRANT	650.507.9844	13/01/08	SH_CLERK	5500
3	200	Jennifer	Whalen	JWHALEN	515.123.4444	17/09/03	AD_ASST	6000
4	201	Michael	Hartstein	MHARTSTE	515.123.5555	17/02/04	MK_MAN	15000
5	202	Pat	Fay	PFAY	603.123.6666	17/08/05	MK_REP	9000
6	203	Susan	Mavris	SMAVRIS	515.123.7777	07/06/02	HR_REP	9000
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10500
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	16000
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	9000
10	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	40000
11	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	30000
12	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	30000
13	103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/06	IT_PROG	10000

c) 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.

```
SAVEPOINT MODIFICA_TODO;
```

Salida de Script | Resultado de la Consu

Tarea terminada en 0,(

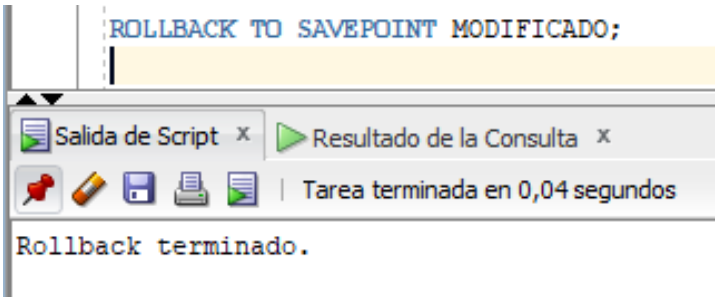
```
Savepoint creado.
```

5. (RA04_f) Se han anulado parcial o totalmente los cambios producidos por una transacción.

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

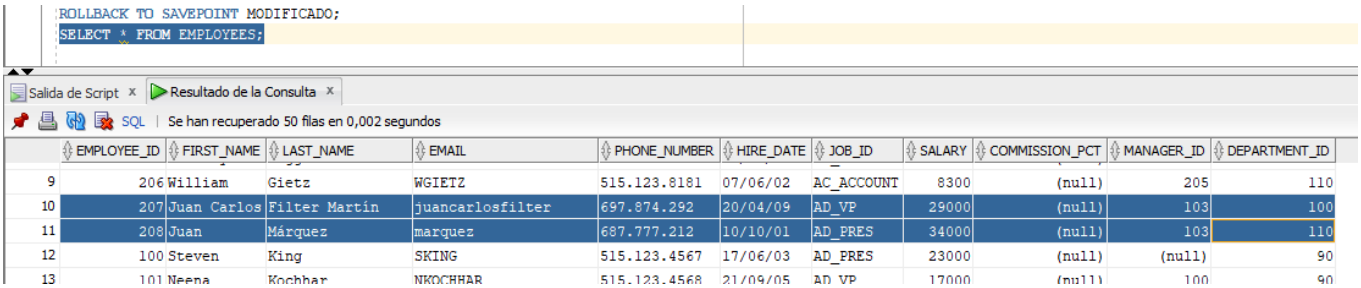
Para volver a un punto de salvado hay que introducir la sentencia:

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)



	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	207	Juan Carlos	Filter Martín	juancarlosfilter	697.874.292	20/04/09	AD_VP	29000	(null)	103	100
11	208	Juan	Márquez	marquez	687.777.212	10/10/01	AD_PRES	34000	(null)	103	110
12	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	23000	(null)	(null)	90
13	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	90

b) 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.

```
ROLLBACK TO SAVEPOINT ELIMINADO;
```

Salida de Script | Resultado de la Consulta

Tarea terminada en 0,043 segundos

```
Error que empieza en la línea: 65 del comando :
ROLLBACK TO SAVEPOINT ELIMINADO
Informe de error -
ORA-01086: nunca se ha establecido el punto de grabación ''ELIMINADO'' en esta sesión o no es válido
01086. 00000 -  "savepoint '%s' never established in this session or is invalid"
*Cause:    An attempt was made to roll back to a savepoint that was never
           established in this session, or was invalid.
*Action:   Try rolling back to the savepoint from the session where it is established.
```

c) 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.

```
ROLLBACK;
```

Salida de Script | Resultado

Tarea term

```
Rollback terminado.
```

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

```
ROLLBACK;
SELECT * FROM EMPLOYEES;
```

Salida de Script | Resultado de la Consulta

Se han recuperado 50 filas en 0,001 segundos

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	198	Donald	OConnell	DOCONNEL	650.507.9833	21/06/07	SH_CLERK	2600	(null)	124	50
2	199	Douglas	Grant	DGRANT	650.507.9844	13/01/08	SH_CLERK	2600	(null)	124	50
3	200	Jennifer	Whalen	JWHALEN	515.123.4444	17/09/03	AD_ASST	4400	(null)	101	10
4	201	Michael	Hartstein	MHARTSTE	515.123.5555	17/02/04	MK_MAN	13000	(null)	100	20
5	202	Pat	Fay	PFAY	603.123.6666	17/08/05	MK_REP	6000	(null)	201	20
6	203	Susan	Mavris	SMAVRIS	515.123.7777	07/06/02	HR_REP	6500	(null)	101	40
7	204	Hermann	Baer	HBAER	515.123.8888	07/06/02	PR_REP	10000	(null)	101	70
8	205	Shelley	Higgins	SHIGGINS	515.123.8080	07/06/02	AC_MGR	12008	(null)	101	110
9	206	William	Gietz	WGIETZ	515.123.8181	07/06/02	AC_ACCOUNT	8300	(null)	205	110
10	100	Steven	King	SKING	515.123.4567	17/06/03	AD_PRES	24000	(null)	(null)	90
11	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/05	AD_VP	17000	(null)	100	90
12	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/01	AD_VP	17000	(null)	100	90
13	103	Alexander	Hunold	AHUNOLD	590.423.4567	03/01/06	IT_PROG	9000	(null)	102	60
14	104	Bruce	Ernst	BERNST	590.423.4568	21/05/07	IT_PROG	6000	(null)	103	60
15	105	David	Austin	DAUSTIN	590.423.4569	25/06/05	IT_PROG	4800	(null)	103	60
16	106	Valli	Pataballa	VPATABAL	590.423.4560	05/02/06	IT_PROG	4800	(null)	103	60
17	107	Diana	Lorentz	DLORENTZ	590.423.5567	07/02/07	IT_PROG	4200	(null)	103	60
18	108	Nancy	Greenberg	NGREENBE	515.124.4569	17/08/02	FI_MGR	12008	(null)	101	100
19	109	Daniel	Faviet	DFAVIET	515.124.4169	16/08/02	FI_ACCOUNT	9000	(null)	108	100
20	110	John	Chen	JCHEN	515.124.4269	28/09/05	FI_ACCOUNT	8200	(null)	108	100
21	111	Ismael	Sciarra	ISCIARRA	515.124.4369	30/09/05	FI_ACCOUNT	7700	(null)	108	100

- O bien podemos hacer una consulta filtrando por ejemplo last_name = ‘Filter Martín’

```
SELECT * FROM EMPLOYEES WHERE LAST_NAME = 'Filter Martín';
```

Salida de Script | Resultado de la Consulta

Todas las Filas Recuperadas: 0 en 0,002 segundos

EMPLOYE...	FIRST_NA...	LAST_NAME	EMAIL	PHONE_N...	HIRE_DATE	JOB_ID	SALARY	COMMISS...	MANAGER...	DEPARTM...

Podemos comprobar que no hay datos.

d) Finaliza y confirma la transacción.

Para finalizar y confirmar la transacción tenemos la sentencian COMMIT

```
COMMIT;
```

Salida de Script | Resultado de

Tarea terminad

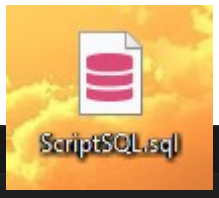
```
Rollback terminado.
Confirmación terminada.
```

Al hacer un COMMIT se guardan permanentemente los cambios realizados.

6. (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.



ScriptSQL.sql

ACTIVIDADES > ScriptSQL.sql

```sql
-- INSERTADO
INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, manager_id, department_id)
VALUES (207,'Juan Carlos', 'Filter Mart►n','juancarlosfilter',
'697.874.292', TO_DATE('20/04/09', 'DD/MM/YY'),'AD_VP', 30000, 103,100);

INSERT INTO EMPLOYEES (employee_id, first_name, last_name, email, phone_number,
hire_date, job_id, salary, manager_id, department_id)
VALUES (208,'Juan', 'M►rquez','marquez',
'687.777.212', TO_DATE('10/10/01', 'DD/MM/YY'),'AD_PRES', 35000, 103,100);

SELECT * FROM EMPLOYEES;
SELECT FIRST_NAME, LAST_NAME, EMAIL, SALARY, DEPARTMENT_ID
FROM EMPLOYEES WHERE department_id = 100;

SAVEPOINT INSERTADO;

 -- MODIFICACI►N

UPDATE employees SET salary = salary - 1000 WHERE salary > 20000;

UPDATE employees SET department_id = 110 WHERE last_name LIKE '%ez';

SELECT * FROM EMPLOYEES;

SAVEPOINT MODIFICADO;

-- ELIMINACI►N

DELETE FROM employees
WHERE first_name = 'Juan Carlos' AND last_name = 'Filter Mart►n';

DELETE FROM employees
WHERE department_id = 110 AND salary >= 30000;
```