

TEMA5

DISEÑO FÍSICO. DML

EJERCICIOS SOLUCIONADOS

Bases de Datos
CFGS DAW

Raquel Torres
raquel.torres@ceedcv.es
Versión:180107.1513

UD05. DISEÑO FÍSICO. DML

1. EJERCICIO 1

El técnico informático de la empresa Laberinto S.L., después de analizar su funcionamiento, ha decidido crear las siguientes tablas:

Tabla Proveedores.

<u>CodProveedor</u> (V10)	NombreProveedor (V30)	CodPostal (V5)
TO342	JUGUETOS, S.A.	45600
MA280	TOYPLAY, S.A.	28005
BA843	CARMELO DIAZ, S.L.	06004
SE391	ARTEAND, S.L.	41400

Tabla ProductosPed

<u>RefeProducto</u> (V10)	NombreProducto V(30)	Precio (Float)
NPP10	NAIPES PETER PARKER	3,00
P3R20	PATINETE 3 RUEDAS	22,50
AFK11	AVION FK20	31,75
PM30	PELUCHE MAYA	15,00
HM12	HOOP MUSICAL	12,80
BB75	BOLA BOOM	22,20

Tabla Pedidos

<u>NumPedido</u> (Int)	Fecha (Date)	Proveedor ▲ (V10)
1	10/06/2013	TO342
2	10/06/2013	MA280
3	12/06/2013	BA843
4	14/06/2013	TO342
5	14/06/2013	MA280

ProductosPedido

<u>NumPedido</u> ▲ (Int)	<u>RefeProducto</u> ▲ (V10)	Cantidad (Int)
1	NPP10	10
1	AFK11	12
2	P3R20	15
3	P3R20	10
3	PM30	20
3	HM12	10
4	AFK11	30
4	BB75	12
5	P3R20	18
5	NPP10	3
5	BB75	5

Crea las tablas indicadas en MySQL y en Oracle e inserta los datos que aparecen en ellas.

(En cada campo se indica el tipo de dato que hay que utilizar al crearlo, por ejemplo (V10) es un

varchar de 10)

1.1 Solución

Realizamos la creación de las tablas. Esta solución te debe funcionar tanto en MySQL como en Oracle.

```
CREATE TABLE proveedores (  
  CodProveedor varchar(10),  
  NombreProveedor varchar(30),  
  CodPostal varchar(5),  
  PRIMARY KEY (CodProveedor)  
);  
  
CREATE TABLE productosped (  
  RefeProducto varchar(10),  
  NombreProducto varchar(30),  
  Prefio float,  
  PRIMARY KEY (RefeProducto)  
);  
  
CREATE TABLE pedidos (  
  NumPedido int,  
  Fecha date,  
  Proveedor varchar(10),  
  PRIMARY KEY (NumPedido),  
  FOREIGN KEY (Proveedor) REFERENCES proveedores (CodProveedor)  
);  
  
CREATE TABLE productospedido (  
  NumPedido int,  
  RefeProducto varchar(10),  
  Cantidad int,  
  PRIMARY KEY (NumPedido, RefeProducto),  
  FOREIGN KEY (NumPedido) REFERENCES pedidos (NumPedido),  
  FOREIGN KEY (RefeProducto) REFERENCES productosped (RefeProducto)  
);
```

Ahora vamos a insertar algunos registros en cada tabla, el resto lo dejo para que practiques tú.

Insertamos datos en la tabla *proveedores*.

```
mysql> insert into proveedores
-> values ('T0342','JUGUETOS, S.A.','45600');
Query OK, 1 row affected (0.03 sec)

mysql> insert into proveedores
-> values ('MA280','TOYPLAY, S.A.','28005');
Query OK, 1 row affected (0.03 sec)

mysql> SELECT * FROM proveedores;
+-----+-----+-----+
| CodProveedor | NombreProveedor | CodPostal |
+-----+-----+-----+
| MA280       | TOYPLAY, S.A.   | 28005     |
| T0342       | JUGUETOS, S.A.  | 45600     |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Insertamos datos en la tabla *productosped*.

```
mysql> insert into productosped
-> values ('NPP10','NAIPES PETER PARKER',3.0);
Query OK, 1 row affected (0.06 sec)

mysql> insert into productosped
-> values ('P3R20','PATINETE 3 RUEDAS',22.50);
Query OK, 1 row affected (0.05 sec)

mysql> insert into productosped
-> values ('AFK11','AVION FK20',31.75);
Query OK, 1 row affected (0.03 sec)
```

Es la primera vez que insertamos datos en un campo *float*, debes tener en cuenta que como es un número va sin comillas y además el separador decimal es el punto (fíjate que si utilizásemos la coma como separador decimal, el sistema interpretaría que le estamos dando dos valores en lugar de uno con parte decimal pues el símbolo que empleamos para separar los campos es la coma).

```
mysql> SELECT * FROM productosped;
+-----+-----+-----+
| RefeProducto | NombreProducto   | Prefio |
+-----+-----+-----+
| AFK11       | AVION FK20       | 31.75  |
| NPP10       | NAIPES PETER PARKER | 3      |
| P3R20       | PATINETE 3 RUEDAS | 22.5   |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

Insertamos datos en la tabla *pedidos*. Recuerda que los campos tipo fecha son diferentes en MySQL y Oracle, aquí el ejemplo se ha realizado sobre MySQL, en Oracle hay que utilizar `TO_DATE('10-06-2013','DD-MM-YYYY')`.

Espero que también hayas observado que los campos de tipo entero también van sin comillas.

```
mysql> insert into pedidos
  -> values (1,'2013-06-10','T0342');
Query OK, 1 row affected (0.05 sec)

mysql> insert into pedidos
  -> values (2,'2013-06-10','MA280');
Query OK, 1 row affected (0.06 sec)

mysql> SELECT * FROM pedidos;
+-----+-----+-----+
| NumPedido | Fecha       | Proveedor |
+-----+-----+-----+
|          1 | 2013-06-10 | T0342     |
|          2 | 2013-06-10 | MA280     |
+-----+-----+-----+
2 rows in set (0.00 sec)
```

Por último insertamos datos en la tabla *productospedido* y mostramos el resultado.

```
mysql> source c:\src\laberinto.sql
Query OK, 0 rows affected (0.13 sec)

mysql> insert into productospedido
  -> values (1,'NPP10',10);
Query OK, 1 row affected (0.03 sec)

mysql> insert into productospedido
  -> values(1,'AFK11',12);
Query OK, 1 row affected (0.06 sec)

mysql> insert into productospedido
  -> values(2,'P3R20',15);
Query OK, 1 row affected (0.06 sec)

mysql> select * from productospedido;
+-----+-----+-----+
| NumPedido | RefeProducto | Cantidad |
+-----+-----+-----+
|          1 | AFK11        |        12 |
|          1 | NPP10        |        10 |
|          2 | P3R20        |        15 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

2. EJERCICIO 2

Recordemos la tabla *departamentos* que hemos creado en la teoría:

```
SQL> select * from departamentos;

CODDPTO  NOMBRE                UBICACION
-----
INF      Informática           Planta sótano U3
ADM      Administración        Planta quinta U2
COM      Comercial             Planta tercera U3
CONT     Contabilidad          Planta quinta U1
MKT      Marketing             Planta cuarta U5
ALM      Almacén               Planta baja U1

6 rows selected.
```

Vamos a crear ahora la tabla de *Empleados* en Oracle y MySQL.

DNI	Nombre	Especialidad	FechaAlta	Dpto ▲
12345678A	Alberto Gil	Contable	10/12/2010	CONT
23456789B	Mariano Sanz	Informática	04/10/2011	INF
34567890C	Iván Gómez	Ventas	20/07/2012	COM
45678901D	Ana Silván	Informática	25/11/2012	INF
56789012E	María Cuadrado	Ventas	02/04/2013	COM
67890123A	Roberto Milán	Logística	05/02/2010	ALM

1. Crear la tabla *Empleados* conservando la integridad referencial (ON DELETE CASCADE, ON UPDATE CASCADE).
2. Insertar los datos de la imagen en la nueva tabla *Empleados* que acabamos de crear. No introduzcas el último registro.
3. Borrar el departamento de Marketing de la tabla *Departamentos*.
4. Insertar ahora el último registro de la tabla *Empleados*, pero en el dpto. De Marketing. ¿Qué ocurre? Razona la respuesta.
5. Insertar ahora el último registro de la tabla *Empleados* tal y como aparece.
6. Eliminar el departamento Comercial (COM) y comprobar que se cumple la integridad referencial, es decir, todos los empleados de ese departamento deben ser eliminados automáticamente.
7. Modificar el código del departamento de informática (INF) por (IT). Comprobar en MySQL que se cumple la cláusula ON UPDATE CASCADE. ¿Se te ocurre alguna forma de realizar esta modificación en Oracle?

2.1 Solución

2.1.1 Crear la tabla *Empleados* conservando la integridad referencial

MySQL.

La instrucción para crear la tabla será:

```
CREATE TABLE empleados(
  dni VARCHAR(10) ,
  nombre VARCHAR(30) ,
  especialidad VARCHAR(25) ,
  fechaalta DATE ,
  dpto VARCHAR(10) ,
  PRIMARY KEY (dni) ,
  FOREIGN KEY (dpto) REFERENCES departamentos(CodDpto)
  ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB;
```

Para crear esta tabla escribiremos el contenido en un documento de texto y ejecutaremos el script correspondiente.

```
mysql> source c:\src\empleados.sql
Query OK, 0 rows affected (0.08 sec)

mysql> desc empleados;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| dni   | varchar(10) | NO | PRI | NULL | |
| nombre | varchar(30) | YES | | NULL | |
| especialidad | varchar(25) | YES | | NULL | |
| fechaalta | date | YES | | NULL | |
| dpto  | varchar(10) | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.02 sec)
```

Oracle.

Recuerda que en Oracle no podemos utilizar la cláusula ON UPDATE, por ello la instrucción a utilizar en Oracle será:

```
CREATE TABLE empleados(
    dni VARCHAR(10),
    nombre VARCHAR(30),
    especialidad VARCHAR(25),
    fechaalta DATE,
    dpto VARCHAR(10),
    PRIMARY KEY (dni),
    FOREIGN KEY (dpto) REFERENCES departamentos(CodDpto)
    ON DELETE CASCADE
);
```

Como antes lo escribiremos en un archivo de texto y ejecutaremos el script. Hasta ahora hemos ejecutado el Script con @ que es el modo abreviado, pero también se pueden ejecutar con START seguido el nombre del script, como en este ejemplo.

```
SQL> START C:\SRC\EMPLEADOS.SQL

Table created.
```

Comprobamos el resultado.

```
SQL> DESC EMPLEADOS;
Name                               Null?    Type
-----
DNI                                NOT NULL VARCHAR2(10)
NOMBRE                             VARCHAR2(30)
ESPECIALIDAD                       VARCHAR2(25)
FECHAALTA                          DATE
DPTO                                VARCHAR2(10)
```

2.1.2 Insertar los datos de la imagen en la nueva tabla *Empleados* que acabamos de crear. No introduzcas el último registro.

Vamos a crear uno a mano y después crearemos un script para los cuatro siguientes, dejando el último, tal y como nos pide el enunciado:

MySQL.

MySQL almacena los campo de tipo fecha con el formato aaaa-mm-dd, es decir: año, mes, día. Para asignar una fecha debemos colocarla en este formato y entre comillas.

```
mysql> insert into empleados
-> values ('12345678A','Alberto Gil','Contable','2010-12-10','CONT');
Query OK, 1 row affected (0.06 sec)

mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni      | nombre    | especialidad | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil | Contable     | 2010-12-10 | CONT |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Como puedes observar el registro ha sido añadido.

Oracle.

Para insertar fechas en Oracle utilizaremos la función *TO_DATE(fecha_string,fecha_formato)* tal y como nos indicaba el enunciado:

```
SQL> INSERT INTO empleados
2 values ('12345678A','Alberto Gil','Contable',
3 TO_DATE('10-12-2010','DD-MM-YYYY'),'CONT');

1 row created.

SQL> select * from empleados;

DNI      NOMBRE      ESPECIALIDAD      FECHAALT
-----
DPTO
-----
12345678A  Alberto Gil      Contable          10/12/10
CONT
```

Vamos a añadir ahora el resto de registros menos el último. Para ello crearemos los scripts correspondientes para cada base de datos.

MySQL.

```
INSERT INTO empleados values
('23456789B','Mariano Sanz','Informática',TO_DATE('04-10-2011','DD-MM-YYYY'),'INF');
INSERT INTO empleados
values ('34567890C','Iván Gómez','Ventas',TO_DATE('20-07-2012','DD-MM-YYYY'),'COM');
INSERT INTO empleados
values ('45678901D','Ana Silván','Informática',TO_DATE('25-11-2012','DD-MM-YYYY'),'INF');
INSERT INTO empleados
values ('56789012E','María Cuadrado','Ventas',TO_DATE('02-04-2013','DD-MM-YYYY'),'COM');
```


Ejecutamos el script :

```
mysql> source c:\src\ins_empleados.sql
Query OK, 1 row affected (0.05 sec)

Query OK, 1 row affected (0.03 sec)

Query OK, 1 row affected (0.05 sec)

Query OK, 1 row affected (0.03 sec)
```

Comprobamos el resultado.

```
mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni      | nombre      | especialidad | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil | Contable     | 2010-12-10 | CONT |
| 23456789B | Mariano Sanz | Informática  | 2011-10-04 | INF  |
| 34567890C | Iván Gómez  | Ventas       | 2012-07-20 | COM  |
| 45678901D | Ana Silván  | Informática  | 2012-11-25 | INF  |
| 56789012E | María Cuadrado | Ventas       | 2013-04-02 | COM  |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Oracle.

Creamos un script con el siguiente contenido.

```
INSERT INTO empleados values
('23456789B', 'Mariano Sanz', 'Informática', TO_DATE('04-10-2011', 'DD-MM-YYYY'), 'INF');
INSERT INTO empleados
values ('34567890C', 'Iván Gómez', 'Ventas', TO_DATE('20-07-2012', 'DD-MM-YYYY'), 'COM');
INSERT INTO empleados
values ('45678901D', 'Ana Silván', 'Informática', TO_DATE('25-11-2012', 'DD-MM-YYYY'), 'INF');
INSERT INTO empleados
values ('56789012E', 'María Cuadrado', 'Ventas', TO_DATE('02-04-2013', 'DD-MM-YYYY'), 'COM');
```

Lo ejecutamos y comprobamos el resultado.

```
SQL> @ c:\src\ins_empleados.sql
1 row created.

1 row created.

1 row created.

1 row created.
SQL> select * from empleados;
DNI      NOMBRE      ESPECIALIDAD      FECHAALT DPT
-----
12345678A Alberto Gil      Contable          10/12/10 CON
23456789B Mariano Sanz     Informática       04/10/11 INF
34567890C Iván Gómez      Ventas            20/07/12 COM
45678901D Ana Silván      Informática       25/11/12 INF
56789012E María Cuadrado   Ventas            02/04/13 COM
```

2.1.3 Borrar el departamento de Marketing de la tabla *Departamentos*.

MySQL.

```
mysql> delete from departamentos
      -> where CodDpto = 'MKT';
Query OK, 1 row affected (0.03 sec)
```

Comprobamos el resultado, el departamento de Marketing ha sido eliminado.

```
mysql> select * from departamentos;
+-----+-----+-----+
| CodDpto | Nombre          | Ubicacion          |
+-----+-----+-----+
| ADM     | Administración  | Planta quinta U2   |
| ALM     | Almacén         | Planta baja U1     |
| COM     | Comercial       | Planta tercera U3   |
| CONT    | Contabilidad    | Planta quinta U1    |
| INF     | Informática     | Planta sótano U3    |
+-----+-----+-----+
5 rows in set (0.00 sec)
```

Oracle.

```
SQL> delete from departamentos
      2  where CodDpto = 'MKT';

1 row deleted.
```

Comprobamos el resultado, el departamento de Marketing ha sido eliminado.

```
SQL> select * from departamentos;

CODDPTO      NOMBRE                UBICACION
-----
INF          Informática           Planta sótano U3
ADM          Administración       Planta quinta U2
COM          Comercial             Planta tercera U3
CONT         Contabilidad          Planta quinta U1
ALM          Almacén               Planta baja U1
```

2.1.4 Insertar ahora el último registro de la tabla *Empleados*, pero en el dpto. De *Marketing*. ¿Qué ocurre? Razona la respuesta.

La inserción del registro se realizará con la siguiente instrucción.

```
INSERT INTO empleados
values ('67890123F','Roberto Milán','Logística','2010-05-02','MKT');
```

¿Qué crees que va a ocurrir?

```
mysql> insert into empleados
      -> values ('67890123F','Roberto Milán','Logística','2010-05-02','MKT');
ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint f
ails ('pruebas_curso/empleados', CONSTRAINT `empleados_ibfk_1` FOREIGN KEY ('dpt
o') REFERENCES `departamentos` ('CodDpto') ON DELETE CASCADE ON UPDATE CASCADE)
mysql>
```

Como puedes ver en la imagen ocurre un error. Recuerda que hemos declarado el campo Dpto de la tabla Empleados como una FOREIGN KEY (clave foránea o extranjera) que se corresponde con el campo CodDpto en la tabla Departamentos.

Eso quiere decir, que si el registro tiene un valor en el campo Dpto ese valor debe existir en la tabla Departamentos en el campo CodDpto.

Sin embargo, tal como hemos comentado no existe ningún registro con el valor 'MKT' y eso es lo que provoca el error que se muestra en la imagen.

Oracle.

Para hacer el insert en Oracle utilizaremos la instrucción:

```
INSERT INTO empleados
values ('67890123F','Roberto Milán','Logística',
TO_DATE('05-02-2010','DD-MM-YYYY'),'MKT');
```

¿Crees que en Oracle también ocurrirá un error?

```
SQL> insert into empleados
2 values ('67890123F','Roberto Milán','Logística',
3 TO_DATE('05-02-2010','DD-MM-YYYY'),'MKT');
insert into empleados
*
ERROR at line 1:
ORA-02291: integrity constraint (USUARIO_PRUEBA.SYS_C007005) violated - parent key not found
```

Por supuesto, pues la clave foránea o extranjera se debe comportar de la misma forma independientemente de la base de datos que estemos utilizando. Puedes ver el error que muestra indicando que no se ha encontrado la clave correspondiente.

2.1.5 Insertar ahora el último registro de la tabla *Empleados* tal y como aparece.

MySQL.

```
mysql> INSERT INTO empleados
-> values ('67890123F','Roberto Milán','Logística','2010-05-02','ALM');
Query OK, 1 row affected (0.05 sec)

mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni   | nombre      | especialidad | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil | Contable     | 2010-12-10 | CONT |
| 23456789B | Mariano Sanz | Informática  | 2011-10-04 | INF  |
| 34567890C | Iván Gómez  | Ventas       | 2012-07-20 | COM  |
| 45678901D | Ana Silván  | Informática  | 2012-11-25 | INF  |
| 56789012E | María Cuadrado | Ventas       | 2013-04-02 | COM  |
| 67890123F | Roberto Milán | Logística    | 2010-05-02 | ALM  |
+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

Oracle.

```
SQL> insert into empleados
  2 values ('67890123F','Roberto Milán','Logística',
  3 TO_DATE('05-02-2010','DD-MM-YYYY'),'ALM');
1 row created.
```

```
SQL> select * from empleados;
```

DNI	NOMBRE	ESPECIALIDAD	FECHAALT	DPT
12345678A	Alberto Gil	Contable	10/12/10	COM
23456789B	Mariano Sanz	Informática	04/10/11	INF
34567890C	Iván Gómez	Ventas	20/07/12	COM
45678901D	Ana Silván	Informática	25/11/12	INF
56789012E	María Cuadrado	Ventas	02/04/13	COM
67890123F	Roberto Milán	Logística	05/02/10	ALM

6 rows selected.

2.1.6 Eliminar el departamento Comercial (COM) y comprobar que se cumple la integridad referencial, es decir, todos los empleados de ese departamento deben ser eliminados automáticamente.

Vamos a comprobar el funcionamiento de la integridad referencial. Teóricamente al establecer ON DELETE CASCADE estamos indicando que en caso de eliminar un departamento, todos los empleados de ese departamento (asociados por la clave foránea creada con el campo Dpto) deben ser eliminados automáticamente. ¿Funcionará?

MySQL.

Los datos de los que partimos son:

```
mysql> select * from departamentos;
```

CodDpto	Nombre	Ubicacion
ADM	Administración	Planta quinta U2
ALM	Almacén	Planta baja U1
COM	Comercial	Planta tercera U3
CONT	Contabilidad	Planta quinta U1
INF	Informática	Planta sótano U3

5 rows in set (0.00 sec)

```
mysql> select * from empleados;
```

dni	nombre	especialidad	fechaalta	dpto
12345678A	Alberto Gil	Contable	2010-12-10	CONT
23456789B	Mariano Sanz	Informática	2011-10-04	INF
34567890C	Iván Gómez	Ventas	2012-07-20	COM
45678901D	Ana Silván	Informática	2012-11-25	INF
56789012E	María Cuadrado	Ventas	2013-04-02	COM
67890123F	Roberto Milán	Logística	2010-05-02	ALM

6 rows in set (0.00 sec)

Ahora vamos a eliminar el departamento Comercial (COM) y comprobaremos el resultado.

```
mysql> delete from departamentos where CodDpto='COM';
Query OK, 1 row affected (0.09 sec)

mysql> select * from departamentos;
+-----+-----+-----+
| CodDpto | Nombre          | Ubicacion          |
+-----+-----+-----+
| ADM     | Administración  | Planta quinta U2   |
| ALM     | Almacén         | Planta baja U1     |
| CONT    | Contabilidad    | Planta quinta U1   |
| INF     | Informática     | Planta sótano U3   |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni      | nombre          | especialidad       | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil     | Contable           | 2010-12-10 | CONT |
| 23456789B | Mariano Sanz    | Informática        | 2011-10-04 | INF  |
| 45678901D | Ana Silván     | Informática        | 2012-11-25 | INF  |
| 67890123F | Roberto Milán   | Logística          | 2010-05-02 | ALM  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Como puedes comprobar al eliminar el departamento Comercial también se han eliminado los empleados que pertenecían a dicho departamento tal como era de esperar al haber establecido el ON DELETE CASCADE.

Oracle.

Realizamos la misma operación en Oracle para ver si el resultado es el mismo.

```
SQL> select * from departamentos;
CODDPDO      NOMBRE          UBICACION
-----
COM          Comercial       Planta tercera U3
CONT         Contabilidad    Planta quinta U1
INF          Informática     Planta sótano U3
ADM          Administración  Planta quinta U2
ALM          Almacén        Planta baja U1

SQL> select * from empleados;
DNI      NOMBRE          ESPECIALIDAD       FECHAALT  DPT
-----
12345678A  Alberto Gil     Contable           10/12/10  CON
23456789B  Mariano Sanz    Informática        04/10/11  INF
34567890C  Iván Gómez     Ventas             20/07/12  COM
45678901D  Ana Silván     Informática        25/11/12  INF
56789012E  María Cuadrado  Ventas             02/04/13  COM
67890123F  Roberto Milán   Logística          05/02/10  ALM

6 rows selected.
```

Borramos el departamento.

```
SQL> delete from departamentos where CodDpto='COM';
1 row deleted.
SQL> select * from departamentos;
CODDPTO      NOMBRE                                UBICACION
-----
CONT         Contabilidad                          Planta quinta U1
INF          Informática                          Planta sótano U3
ADM          Administración                       Planta quinta U2
ALM          Almacén                              Planta baja U1

SQL> select * from empleados;
DNI          NOMBRE                                ESPECIALIDAD      FECHAALT  DPT
-----
12345678A    Alberto Gil                           Contable          10/12/10  COM
23456789B    Mariano Sanz                          Informática       04/10/11  INF
45678901D    Ana Silván                            Informática       25/11/12  INF
67890123F    Roberto Milán                         Logística        05/02/10  ALM
```

Igual que en el caso anterior puedes comprobar que ha funcionado correctamente habiendo desaparecido los empleados correspondientes al departamento Comercial.

- 2.1.7 Modificar el código del departamento de informática (INF) por (IT). Comprobar en MySQL que se cumple la cláusula ON UPDATE CASCADE. ¿Se te ocurre alguna forma de realizar esta modificación en Oracle?

Los datos de los que partimos son:

```
mysql> select * from departamentos;
+-----+-----+-----+
| CodDpto | Nombre          | Ubicacion        |
+-----+-----+-----+
| ADM     | Administración  | Planta quinta U2 |
| ALM     | Almacén         | Planta baja U1   |
| CONT    | Contabilidad    | Planta quinta U1 |
| INF     | Informática     | Planta sótano U3 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni    | nombre          | especialidad     | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil    | Contable        | 2010-12-10 | CONT |
| 23456789B | Mariano Sanz   | Informática     | 2011-10-04 | INF  |
| 45678901D | Ana Silván     | Informática     | 2012-11-25 | INF  |
| 67890123F | Roberto Milán  | Logística       | 2010-05-02 | ALM  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Realizamos la actualización.

```
mysql> update departamentos set CodDpto='IT' where CodDpto='INF';
Query OK, 1 row affected (0.08 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Los datos modificados son:

```
mysql> select * from departamentos;
+-----+-----+-----+
| CodDpto | Nombre          | Ubicacion          |
+-----+-----+-----+
| ADM     | Administración  | Planta quinta U2   |
| ALM     | Almacén         | Planta baja U1     |
| CONT    | Contabilidad    | Planta quinta U1   |
| IT      | Informática     | Planta sótano U3   |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql> select * from empleados;
+-----+-----+-----+-----+-----+
| dni      | nombre          | especialidad       | fechaalta | dpto |
+-----+-----+-----+-----+-----+
| 12345678A | Alberto Gil     | Contable           | 2010-12-10 | CONT |
| 23456789B | Mariano Sanz    | Informática        | 2011-10-04 | IT   |
| 45678901D | Ana Silván     | Informática        | 2012-11-25 | IT   |
| 67890123F | Roberto Milán   | Logística          | 2010-05-02 | ALM  |
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

Podemos observar que los empleados que pertenecían a INF ahora están en IT, es decir, la modificación se ha propagado correctamente.

¿Qué ocurre si lo intentamos hacer en ORACLE?

Vamos a probar, recuerda que en Oracle no tenemos ON UPDATE CASCADE.

```
SQL> update departamentos set coddpto='IT' where coddpto='INF';
update departamentos set coddpto='IT' where coddpto='INF'
*
ERROR at line 1:
ORA-02292: integrity constraint (USUARIO_PRUEBA.SYS_C007005) violated - child
record found
```

Nos aparece un error y nos dice que no puede modificar la clave principal porque perdería la integridad al haber registros relacionados que contienen esa clave.

Para hacerlo de forma manual, primero hay que poner a nulos los registros que tienen ese valor (INF) en el campo *dpto* de la tabla *empleados*.

Después ya podremos cambiar la clave INF por IT.

Por último debemos volver a asignar a los registros que pusimos a nulos el nuevo valor IT.

(Lo hacemos para mantener las dos bases de datos (MySQL y Oracle) con la misma información.)

Ponemos a nulos los registros del departamento INF.

```
SQL> update empleados set dpto=NULL where dpto='INF';  
2 rows updated.
```

Modificamos el valor de la clave del departamento de informática.

```
SQL> update departamentos set coddpto='IT' where coddpto='INF';  
1 row updated.
```

Cambiamos los registros de empleados donde hemos puesto *dpto* a nulos al nuevo valor IT.

```
SQL> update empleados set dpto='IT' where dpto is NULL;  
2 rows updated.
```

Es importante que os fijéis en cómo se hace la comparación en esta última instrucción, no hemos puesto *dpto=NULL*, hemos puesto *dpto is NULL*. El valor nulo tiene un tratamiento especial (ya comenzamos a verlo en la unidad anterior, cuando definíamos la restricción VNN) y lo estudiaremos con más detalle en el próximo tema.

Pero de momento, que sepáis que para conocer si el contenido de un campo es nulo o no, se hace con *is NULL*.

En cualquier caso, este ejercicio es simplemente para que veáis como afecta la integridad referencial, pero en el mundo real muy pocas veces se efectúan cambios en las claves principales.

Es raro que a una persona le modifiquen el DNI, que a un coche le cambien la matrícula, que un libro cambie su ISBN, que a un alumno le cambien el número de expediente, que un trabajador cambie su número de seguridad social, etc. Si bien no es imposible, suele ser poco probable que se cambien los valores de las claves principales de los registros.