

Transacciones

Considera las siguientes secuencias de sentencias SQL y contesta a las preguntas **sin ejecutar nada**. Una vez hayas contestado a las preguntas ejecuta el código para comprobar tus respuestas con los resultados obtenidos.

■ Ejercicio 1

```
1 CREATE TABLE Enrolment (
2   ID CHAR(10) PRIMARY KEY,
3   Subject CHAR(10),
4   Credits NUMBER(2,0)
5 );
6 SAVEPOINT paso_uno;
7 INSERT INTO ENROLMENT VALUES ('123456789X', 'DDBB', 6);
8 SAVEPOINT paso_dos;
9 UPDATE Enrolment SET Credits= Credits+1 WHERE ID = '123456789X' and Subject = '
  DDBB';
10 ROLLBACK TO SAVEPOINT paso_dos;
11 UPDATE Enrolment SET Credits= Credits+2 WHERE ID = '123456789X' and Subject = '
  DDBB';
12 INSERT INTO Enrolment VALUES ('123456789X', 'DDBB', 12);
13 UPDATE Enrolment SET Credits = Credits+3 WHERE ID = '123456789X' and Subject='
  DDBB';
14 SAVEPOINT paso_tres;
15 COMMIT;
16 CREATE TABLE AcademicRecord(
17   ID VARCHAR2(20) PRIMARY KEY,
18   Total NUMBER(5)
19 );
20 INSERT INTO AcademicRecord VALUES('00000000P', 45);
21 ROLLBACK TO SAVEPOINT paso_tres;
22 SELECT * FROM AcademicRecord where ID = '00000000P';
23 ROLLBACK;
24 SELECT * FROM AcademicRecord where ID = '00000000P';
25 COMMIT;
```

1. ¿Hay alguna sentencia de control de transacciones que no sea necesaria (aunque sea correcta)? ¿Cuáles y por qué?
2. ¿Qué sentencias producen errores durante la ejecución del código SQL? ¿Por qué?
3. Muestra el resultado de las sentencias en las líneas 22 y 23.
4. ¿Cuántas transacciones hay en el código? Indica la primera y última sentencia de cada transacción.

■ Ejercicio 2

```
1 SAVEPOINT paso_uno;
2 CREATE TABLE Sales(
3   Title VARCHAR2(25) PRIMARY KEY,
4   TicketsSold NUMBER(3,0)
5 );
6 INSERT INTO Sales VALUES ('My Fair Lady', 200);
7 SAVEPOINT paso_dos;
8 UPDATE Sales SET TicketsSold = TicketsSold + 100 WHERE Title = 'My Fair Lady';
9 COMMIT;
10 UPDATE Sales SET TicketsSold = TicketsSold + 200 WHERE Title = 'My Fair Lady';
```

```

11 ROLLBACK;
12 INSERT INTO Sales VALUES ('My Fair Lady', 315);
13 UPDATE SALES SET TicketsSold = TicketsSold + 300 WHERE Title = 'Hamlet';
14 COMMIT;
15 CREATE TABLE TopSales(
16     Tfilm VARCHAR2(70),
17     Total NUMBER(5)
18 );
19 INSERT INTO TopSales VALUES ('Breakfast at Tiffanys', 100);
20 ROLLBACK TO SAVEPOINT paso_tres;
21 SELECT * FROM TopSales WHERE Tfilm = 'Breakfast at Tiffanys';
22 SELECT * FROM Sales;
23 COMMIT;

```

1. ¿Hay alguna sentencia de control de transacciones que no sea necesaria (aunque sea correcta)? ¿Cuáles y por qué?
2. ¿Qué sentencias producen errores durante la ejecución del código SQL? ¿Por qué?
3. Muestra el resultado de las sentencias en las líneas 21 y 22.
4. ¿Cuántas transacciones hay en el código? Indica la primera y última sentencia de cada transacción.

■ Ejercicio 3

```

1 SAVEPOINT paso_uno;
2 CREATE TABLE P1(id INT PRIMARY KEY, nE INT DEFAULT 0);
3 ROLLBACK;
4 INSERT INTO P1 VALUES (101, 100);
5 SAVEPOINT paso_uno;
6 INSERT INTO P1 VALUES (102, 200);
7 UPDATE P1 SET nE = nE + 3000;
8 ROLLBACK TO SAVEPOINT paso_uno;
9 CREATE TABLE Aux(nombre VARCHAR2(10), edad INTEGER);
10 INSERT INTO Aux VALUES ('Marta', 25);
11 SAVEPOINT paso_dos;
12 DROP TABLE P1;
13 UPDATE Aux SET edad = edad + 2;
14 ROLLBACK TO SAVEPOINT paso_dos;

```

1. Indica las instrucciones de control de transacciones que son innecesarias y las que provocan error, explicando el motivo.
2. ¿Qué datos y qué tablas quedan al final de la ejecución del *script*?
3. ¿Cuántas transacciones se han ejecutado? Indica el inicio y el fin de cada una de ellas usando los números de línea.

■ Ejercicio 4

```

1 SAVEPOINT paso_uno;
2 CREATE TABLE t(a integer);
3 SAVEPOINT paso_dos;
4 ROLLBACK TO SAVEPOINT paso_uno;
5 CREATE TABLE t(a integer);
6 INSERT INTO t VALUES (0);
7 SAVEPOINT paso_tres;
8 UPDATE t SET a=a+1;
9 COMMIT;
10 ROLLBACK TO SAVEPOINT paso_tres;
11 SELECT a FROM t WHERE a > 0;

```

1. ¿Hay instrucciones de control de transacciones que, aun siendo correctas, son innecesarias? ¿Cuáles y por qué?

2. ¿Alguna instrucción provocaría un error en su ejecución? ¿Cuáles y por qué?
3. Indica el resultado de la instrucción **SELECT** de la línea 11.
4. ¿Cuántas transacciones se han ejecutado? Indica el inicio y el fin de cada una de ellas usando los números de línea.

Conexión con una BD de datos relacional desde un programa

Siguiendo el ejemplo de conexión con Oracle que aparecen en los apuntes de la asignatura (*Conceptos Avanzados*), crea un programa Java que:

1. Se conecta a la BD Oracle con vuestro usuario y contraseña.
2. Inserta dos personas nuevas en la BD. Engloba estas modificaciones en una única transacción (posiblemente necesites desactivar el *auto commit* de la conexión).
3. Muestra por pantalla el NIF y nombre de las personas que no han asistido a ningún partido. Deben aparecer las nuevas personas insertadas.
4. Muestra por pantalla el nombre de los clubes que han ganado todos sus partidos como local (y han jugado al menos uno).
5. Muestra por pantalla los datos completos de los clubes financiados por todos los patrocinadores.
6. Elimina de la base de datos los jugadores que has añadido en el primer paso. Trata estas las modificaciones como una única transacción y confírmala.