

Consultas en AR y en SQL - Ejercicios adicionales

1. Sean las relaciones siguientes:

SOCIO (AFICIONADO, VIDEOCLUB)
GUSTA (AFICIONADO, PELICULA)
VIDEOTECA (VIDEOCLUB, PELICULA)

Escribir en AR y en SQL las sentencias necesarias para responder a las preguntas siguientes:

- Videoclubes que disponen de alguna película que le guste al aficionado José Pérez.
- Aficionados que son socios al menos de un videoclub que dispone de alguna película de su gusto.
- Aficionados que no son socios de ningún videoclub donde tengan alguna película de su gusto.
- Videoclubes que disponen tanto de la película A como de la película B (ambas).
- Aficionados que son socios de al menos dos videoclubes que tengan alguna película de su gusto.

2. Sean las relaciones siguientes:

EDITORIALES (EDITORIAL#, NOMBRE, CIUDAD)
LIBROS (LIBRO#, TITULO, AUTOR, AÑO)
LIBRERÍAS (LIBRERIA#, NOMBRE, CIUDAD)
ELL (EDITORIAL#, LIBRO#, LIBRERIA#, CANTIDAD)

Se pide escribir en AR y en SQL las respuestas a las preguntas siguientes:

- Obtener los nombres de las librerías abastecidas por alguna editorial de Rosario.
- Obtener los códigos de las editoriales que suministran a las librerías P1 y P3 (ambas).
- Obtener los códigos de librerías abastecidas por todas las editoriales.

3. Se dispone de una BD que almacena información referente a partidas de caza realizadas en distintos cotos (un coto es un campo privado). La BD está formada por las relaciones siguientes:

GUIAS (COD-GUIA, NOMBRE, DIRECCION, TELEFONO)
COTOS (COD-COTO, NOMBRE, EXTENSION)
ESPECIES (COD-ESPECIE, NOMBRE, FECHA-INICIO-VEDA, FECHA-FIN-VEDA)
PARTIDA (COD-GUIA, COD-COTO, COD-ESPECIE, FECHA, NUMERO_PIEZAS)

Cada partida de caza es liderada por un guía, y es realizada en un solo coto. La duración de cada partida de caza es de un día. En cada partida, se cazan cierta cantidad de piezas. Cada especie tiene un rango de fechas (FECHA-INICIO-VEDA, FECHA-FIN-VEDA) entre las cuales puede ser cazada.

Se pide dar soluciones en AR Y SQL a las siguientes consultas:

- Obtener los nombres y direcciones correspondientes a los guías que no han participado en ninguna partida de caza.
- Obtener los nombres de los guías que, habiendo participado en partidas de caza, no han realizado partidas de caza en los cotos de código C3 ó C7.
- Obtener los nombres de los guías que han realizado partidas de caza en el coto de código C1 y en el coto de código C5.
- Obtener los nombres de los cotos en los que se han hecho partidas de caza para todas las especies.

4. Sean las siguientes relaciones:

ALUMNOS (A#, NOMBRE, GRUPO)
PRACTICAS (P#, CURSO, FECHA)
ENTREGA (A#, P#, NOTA)

Dar respuesta en SQL y AR a las siguientes consultas:

- a) Encontrar los nombres de los alumnos que pertenecen al grupo C ó al grupo F.
- b) Encontrar los nombres de los alumnos que han entregado todas las prácticas de tercer curso.
- c) Encontrar los nombres de los alumnos que han entregado prácticas de segundo y tercer curso (ambas).
- d) Encontrar los nombres de los alumnos que sólo han entregado prácticas de segundo curso.
- e) Encontrar los nombres de los alumnos que están en el mismo grupo que Juan. El legajo de Juan es A25.
- f) Encontrar los nombres de los alumnos que no han realizado ninguna entrega.

Ejemplo de ejercicio de Integridad

Las sentencias de creación de las tablas de la BD *Docencia* son las siguientes:

```
CREATE TABLE DEPTO (cod char(5) PRIMARY KEY, nombre char(100) NOT NULL,  
    director char(5), telefono CHAR(11),  
    FOREIGN KEY director REFERENCES Profesor (cod)  
    ON UPDATE CASCADE ON DELETE NULLIFIES );
```

```
CREATE TABLE PROFESOR (cod char(5) PRIMARY KEY, nombre char(50) NOT NULL,  
    categoría char(3) NOT NULL, depto char(5) NOT NULL, teléfono char(11),  
    FOREIGN KEY depto REFERENCES Depto (cod) ON UPDATE CASCADE);
```

```
CREATE TABLE ASIGNATURA (cod char(5) PRIMARY KEY, nombre char(100) NOT NULL,  
    semestre char(2) NOT NULL, depto char(5) NOT NULL,  
    FOREIGN KEY depto REFERENCES Depto (cod)  
    ON UPDATE CASCADE ON DELETE RESTRICTED );
```

```
CREATE TABLE TAREADOCENTE (codprof char(5), codasig char(5), teoría integer NOT NULL,  
    practica integer NOT NULL, PRIMARY KEY (codprof, codasig),  
    FOREIGN KEY codasig REFERENCES Asignatura (codigo) ON UPDATE CASCADE,  
    FOREIGN KEY codprof REFERENCES Profesor (codigo)  
    ON DELETE CASCADE ON UPDATE CASCADE );
```

Explique coloquialmente (¿se puede hacer?, ¿no se puede hacer?, ¿por qué?) qué ocurre cuando se intenta:

- a) borrar un departamento
- b) cambiar el código de un profesor
- c) borrar una asignatura
- d) borrar un profesor
- e) modificar el código de un profesor en la tabla de tareas docentes