



Bases de Datos 2016-2

Prof. Angélica Caro, 6-10-2016

Ejercicios SQL: Base de Datos Acuario

- Considere las siguientes relaciones de la base de datos Acuario. El atributo subrayado corresponde a la clave primaria de la relación.

especies(sno: integer, nombre_especie: varchar(30), alimento: varchar(40))

tanques(tno: integer, nombre_tanque: varchar(30), color_tanque: varchar(20), volumen: integer)

peces(pno: integer, nombre_peces: varchar(30), color_peces: varchar(20), *tno*: integer, *sno*: integer)

Donde los atributos *tno* y *sno* son claves foráneas que hacen referencia a las claves primarias de las relaciones *tanques* y *especies* respectivamente.

eventos(eno: integer, *pno*: integer, fecha: date)

Donde *pno* es clave foránea y referencia al atributo *pno* de la tabla *peces*.

- Considere además la siguiente instancia de la base de datos:

especies		
sno	nombre	alimento
17	delfin	arenque
22	tiburón	cualquier cosa
74	olomina	gusano
93	ballena	mantequilla de mani
100	pez espada	gusano
120	pez globo	gusano

tanques			
tno	nombre_tanque	color_tanque	volumen
55	charco	verde	300
42	letrina	azul	100
35	laguna	rojo	400
85	letrina	azul	100
38	playa	azul	200
44	laguna	verde	200

peces				
pno	nombre_peces	color_peces	tno	sno
164	charlie	naranja	42	74
347	flipper	negro	35	17
228	killer	blanco	42	22
281	albert	rojo	55	17
119	bonnie	azul	42	22
388	cory	morado	35	93
700	maureen	blanco	44	100
800	beni	rojo	55	17
900	nemo	rojo	44	74
150	vicky	rojo	55	100
160	mati	amarillo	42	100
110	rafa	azul	85	100
222	jimmy	amarillo	38	100
144	bisho	rojo	42	93
125	chris	azul	38	93
183	sable	amarillo	44	93
241	taz	rojo	55	93
300	baltasar	azul	85	100
200	cash	azul	85	100
424	bandido	verde	35	100
454	romo	blanco	85	93

eventos		
eno	pno	fecha
3456	347	2010-01-26
6653	164	2010-05-14
5644	347	2010-05-15
5645	347	2010-05-30
6789	281	2010-04-30
5211	228	2010-08-20
6719	700	2010-10-22
4555	164	2011-11-03
9647	281	2011-12-06
5347	281	2011-01-01



- Exprese en SQL las siguientes consultas:

1. Listar el código y nombre de los tanques que tienen peces de color rojo o peces de color azul (usar operador UNION).

```
SELECT tno,nombre_tanque from tanques
where tno IN ((SELECT tno from peces where color_peces = 'rojo')
UNION (SELECT tno from peces where color_peces = 'azul' ));
```

2. Indicar código y nombre los tanques que contienen tiburones.

```
SELECT tno,nombre_tanque from tanques
where tno IN (SELECT tno from peces where sno
IN (SELECT sno from especies where nombre_especie = 'tiburon' ));
```

3. Indicar código y nombre los tanques que no contienen tiburones (usar operador EXCEPT) .

```
SELECT tno,nombre_tanque from tanques
where tno IN ((SELECT tno from peces)
EXCEPT (SELECT tno from peces where sno
IN (SELECT sno from especies where nombre_especie = 'tiburon' )));
```

4. Listar el código y nombre de los tanques que solo tienen peces de color rojo o peces de color azul.

```
SELECT tno , nombre_tanque from tanques
where tno IN ( ( SELECT tno from peces where color_peces = 'rojo' or color_peces = 'azul' )
EXCEPT (SELECT tno from peces where color_peces <> 'rojo' and color_peces <> 'azul' ) ) ;
```

5. Listar código y nombre de los tanques que contienen peces que comen gusanos y arenque (usar operador INTERSECT) .

```
SELECT tno , nombre_tanque from tanques
where tno IN ( (SELECT tno from peces where sno IN
(SELECT sno from especies where alimento = 'arenque' ) )
INTERSECT (SELECT tno from peces where sno IN
(SELECT sno from especies where alimento = 'gusano' ) ) ) ;
```



6. Encontrar el nombre de las especies que no tienen peces de color morado.

```
SELECT nombre_especie from especies E where NOT EXISTS  
(SELECT * from peces P where color_peces = 'morado' and P.sno = E.sno);
```

7. Listar el nombre de las especies que tienen peces en todos los tanques.

```
SELECT nombre_especie from especies E WHERE NOT EXISTS  
( ( SELECT tno from tanques ) EXCEPT (SELECT tno from peces P where E.sno = P.sno) ) ;
```

8. Listar el código de los peces y el total de participaciones en eventos (usar funciones de agregación).

```
SELECT pno, COUNT(*) from peces natural join eventos group by pno;
```

9. Listar el promedio de volumen de los tanques del acuario.

```
SELECT AVG (volumen) from tanques;
```

10. Listar el nombre y color de los tanques que tiene el volumen mayor.

```
SELECT tno, nombre_tanque from tanques  
where volumen = (SELECT MAX(volumen) FROM tanques);
```

11. Listar el nombre y volumen mínimo, de los tanques que tienen el volumen mínimo del acuario.

```
SELECT nombre_tanque, volumen from tanques  
where volumen = (SELECT MIN(volumen) FROM tanques);
```

12. Listar el nombre del tanque y la cantidad de peces que contiene cada uno.

```
SELECT nombre_tanque, COUNT(*)  
from (peces natural join tanques) group by nombre_tanque, tno;
```

13. Listar el código y nombre de las especies que tienen más peces de color rojo.

```
SELECT sno, nombre_especie  
from (SELECT * from (SELECT sno, COUNT(*) as tot_rojos from peces where  
color_peces ='rojo' group by sno) M Natural Join especies) A  
WHERE A.tot_rojos = (SELECT MAX(tot_rojos) from (SELECT sno, COUNT(*) as  
tot_rojos from peces where color_peces ='rojo' group by sno) B);
```



14. Genere una consulta que obtenga el código y nombre de las especies que no tienen peces.

```
SELECT sno, nombre_especie from especies  
where sno IN (SELECT sno FROM especies EXCEPT SELECT sno FROM peces);
```

15. Cree una consulta que liste las especies que tiene más de 3 peces

```
SELECT sno, nombre_especie FROM especies  
WHERE (SELECT COUNT(*) FROM peces WHERE especies.sno=peces.sno) > 3;
```