**SOLUCIONES ACTIVIDADES ADICIONALES TEMA 10**

Vamos a trabajar con la base de datos *Tauromaquia*, cuyo script debe ejecutar. El esquema relacional de esta base de datos es el que se muestra a continuación:

Toreros (id\_torero, nombre)

Toros (id\_toro, nombre, kilos, orejas, rabos, id\_torero, id\_corrida)

Corridas (id\_corrida, fecha\_corrida, espectadores, televisada, id\_plaza)

Plazas (id\_plaza, nombre, aforo, localidad)

En la figura se muestran subrayadas las claves primarias y mediante flechas las claves ajenas o externas.

En la siguiente tabla se indica por cada uno de los atributos de que constan estas tablas su nombre, descripción y las restricciones que soporta.

|  |  |  |  |
| --- | --- | --- | --- |
| Tabla | **Atributo** | **Descripción** | **Restricciones** |
| Toreros | id\_torero | Nº identificativo del torero | Autonumérico |
| Nombre | Nombre del torero | Obligatorio y único |
| Toros | Id\_toro | Nº identificativo del toro | Autonumérico |
| Nombre | Nombre del toro | Obligatorio y único |
| Kilos | Peso del toro en kilos | Obligatorio. |
| Orejas | Nº de orejas que se le han cortado al toro | Valores posibles: 0 , 1, 2. Por defecto 0. |
| Rabos | Nº rabos que se le han cortado al toro | Valores posibles: 0, 1. Por defecto 0. |
| Id\_torero | Nº identificativo del torero que lo ha toreado |  |
| Id\_corrida | Nº identificativo de la corrida en la que se ha toreado al toro |  |
| Corridas | Id\_corrida | Nº identificativo de la corrida | Autonumérico |
| Fecha\_corrida | Fecha de la corrida | Obligatorio. |
| Espectadores | Nº de espectadores de la corrida |  |
| Televisada | Indica si la corrida ha sido televisada (1) o no (0) | Valores posibles: 0, 1. Por defecto 0. |
| Id\_plaza | Nº identificativo de la plaza en que se celebra la corrida |  |
| Plazas | Id\_plaza | Nº identificativo de la plaza | Autonumérico. |
| Nombre | Nombre de la plaza | Obligatorio |
| Aforo | Nº de espectadores que caben en la plaza |  |
| Localidad | Nombre de la localidad donde está la plaza | Obligatorio |

1. Cree una vista llamada *VCorridas* que almacene por cada corrida de la base de datos, su fecha, el número de espectadores y el nombre, localidad y aforo de la plaza donde se celebró la corrida, así como el número de plazas libres que hubo en la corrida, dato que se calcula restado al aforo de la plaza el número de espectadores de la corrida. Llame a los atributos de la vista *FecCor*, *EspCor*, *NomPla*, *LocPla*, *AfoPla* y *Vacias*, respectivamente.

Create view VCorridas (FecCor, EspCor, NomPla, LocPla, AfoPla, Vacias) as

Select fecha\_corrida, espectadores, nombre, localidad, aforo, aforo-espectadores

From Corridas C join Plazas P where C.id\_plaza = P.id\_plaza;

1. Intente incrementar en 1000 espectadores a través de la vista *VCorridas* el número de espectadores de la corrida celebrada el 17 de abril de 2017.. Escriba la orden SQL necesaria para ello. Si es posible realizar esta operación, indiqué por qué es posible escribiendo todas las condiciones que cumple la vista para que la operación sea posible. En caso de que no sea posible, indique también por qué no es posible.

update VCorridas

set EspCor = EspCor + 1000

where FecCor = '2017-04-17';

Es posible porque:

* Hay una relación 1 a 1 entre los registros de la vista VCorridas y los de la tabla subyacente Corridas.
* No se usan en la definición de la vista cláusulas group by ni distinct.
* No se usan en la definición de la vista uniones ni reuniones externas.
* El atributo que se modifica no está definido mediante una expresión.

1. Intente modificar a través de la vista *VCorridas* el nº de plazas libres que quedaron en la corrida celebrada el 16 de abril de 2017, de manera que el nuevo valor sea de 1000 plazas. Escriba la orden SQL necesaria para ello. Si es posible realizar esta operación, indiqué por qué es posible escribiendo todas las condiciones que cumple la vista para que la operación sea posible. En caso de que no sea posible, indique también por qué no es posible.

update VCorridas

set Vacias = 1000

where FecCor = '2017-04-16';

No es posible porque el atributo que se modifica (*Vacias*) es una columna derivada, o lo que es lo mismo, está definido mediante una expresión (*aforo*-*espectadores*).

1. Cree una vista llamada *VToros* que almacene por cada toro de más de 500 kg. de la base de datos su identificativo, su nombre y el número de orejas y rabos.

Create view VToros as

Select id\_toro, nombre, orejas, rabos

From Toros

Where kilos > 500;

1. Intente insertar a través de esta vista un nuevo toro en la base de datos con nombre *Chiquito* y al que se le ha cortado solo 1 oreja. Tenga en cuenta que el campo *id\_toro* es autonumérico y que, por tanto, no es necesario asignarle valor en la orden SQL. Escriba la orden SQL necesaria para ello. Si es posible realizar esta operación, indique por qué es posible escribiendo todas las condiciones que cumple la vista para que la operación sea posible. En caso de que no sea posible, indique también por qué no es posible.

insert into VToros (nombre, orejas, rabos)

values ('Chiquito', '1', '0');

No es posible porque hay un atributo de la tabla *Toros* (*kilos*) que no está en la vista y que en la tabla es obligatorio y no tiene asignado valor por defecto.

1. Intente eliminar al toro llamado Malabestia de la base de datos usando la vista *VToros*. Escriba la orden SQL necesaria para ello. Si es posible realizar esta operación, indique por qué es posible escribiendo todas las condiciones que cumple la vista para que la operación sea posible. En caso de que no sea posible, indique también por qué no es posible.

delete from VToros

where nombre = 'Malabestia';

Es posible porque:

* Hay una relación 1 a 1 entre los registros de la vista *VToros* y los de la tabla subyacente *Toros*.
* No se usan en la definición de la vista cláusulas group by ni distinct.
* No se usan en la definición de la vista uniones ni reuniones externas.

1. Lleve a cabo las siguientes operaciones relacionadas con la gestión de usuarios sobre la base de datos *Empresa*:

Depart (dept\_no, dnombre, loc)

Emple (emp\_no, apellido, oficio, dir, fecha\_alt, salario, comision, dept\_no)

1. Cree un usuario llamado *adminempresa* que solo se pueda conectar desde el equipo local con contraseña ‘empresa’. Imponga como restricción a este usuario que no pueda realizar más de 20 actualizaciones por cada hora.

mysql> create user adminempresa@localhost identified by 'empresa'

with max\_updates\_per\_hour 20;

Query OK, 0 rows affected (0.02 sec)

1. Conceda a este usuario la posibilidad de crear, alterar y eliminar tablas en la base de datos *Empresa*. Además, este usuario debe poder conceder sus privilegios a otros usuarios.

mysql> grant create, alter, drop on empresa.\*

-> to adminempresa@localhost

-> with grant option;

Query OK, 0 rows affected (0.00 sec)

1. Conceda a este usuario la posibilidad de consultar, añadir y borrar filas sobre la tabla *Emple*, así como modificar los atributos *oficio*, *salario* y *comision* para empleados en dicha tabla. Visualice sus permisos. A continuación compruebe que puede modificar la comisión de un empleado, pero no el número de departamento en el que trabaja.

mysql> grant select, insert, delete, update (oficio, salario, comision)

-> on empresa.emple

-> to adminempresa@localhost;

Query OK, 0 rows affected (0.13 sec)

mysql> show grants for adminempresa@localhost;

+---------------------------------------------------------------------------------------------------------------------+

| Grants for adminempresa@localhost |

+---------------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'adminempresa'@'localhost' |

| GRANT CREATE, DROP, ALTER ON `empresa`.\* TO 'adminempresa'@'localhost' WITH GRANT OPTION |

| GRANT SELECT, INSERT, UPDATE (comision, salario, oficio), DELETE ON `empresa`.`emple` TO 'adminempresa'@'localhost' |

+---------------------------------------------------------------------------------------------------------------------+

3 rows in set (0.00 sec)

mysql> update emple

-> set comision = 100 where apellido = 'JIMÉNEZ';

Query OK, 1 row affected (0.61 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update emple

-> set dept\_no = 30 where apellido= 'JIMÉNEZ';

ERROR 1143 (42000): UPDATE command denied to user 'adminempresa'@'localhost' for column 'dept\_no' in table 'emple'

1. Conceda a este usuario la posibilidad de consultar el contenido de la tabla *Depart*. Compruebe que puede consultar dicha tabla, pero no añadir nuevos departamentos.

mysql> grant select on empresa.Depart

-> to adminempresa@localhost;

Query OK, 0 rows affected (0.00 sec)

mysql> select \* from depart;

+---------+---------------+-----------+

| dept\_no | dnombre | loc |

+---------+---------------+-----------+

| 10 | CONTABILIDAD | SEVILLA |

| 20 | INVESTIGACIÓN | MADRID |

| 30 | VENTAS | BARCELONA |

| 40 | PRODUCCIÓN | BILBAO |

+---------+---------------+-----------+

4 rows in set (0.03 sec)

mysql> insert into Depart values (50, 'PRUEBA', 'BURGOS');

ERROR 1142 (42000): INSERT command denied to user 'adminempresa'@'localhost' for table 'depart'

1. Conceda a este usuario la posibilidad de crear vistas y rutinas en la base de datos *Empresa*.

mysql> grant create view, create routine

-> on Empresa.\*

-> to adminempresa@localhost;

Query OK, 0 rows affected (0.00 sec)

1. Retire a este usuario la posibilidad de consultar la tabla *Depart*. Visualice sus permisos. Compruebe que en efecto no puede consultar dicha tabla.

mysql> revoke select on empresa.Depart

-> from adminempresa@localhost;

Query OK, 0 rows affected (0.00 sec)

mysql> show grants for adminempresa@localhost;

+-----------------------------------------------------------------------------------------------------------------------+

| Grants for adminempresa@localhost |

+-----------------------------------------------------------------------------------------------------------------------+

| GRANT USAGE ON \*.\* TO 'adminempresa'@'localhost' |

| GRANT CREATE, DROP, ALTER, CREATE VIEW, CREATE ROUTINE ON `empresa`.\* TO 'adminempresa'@'localhost' WITH GRANT OPTION |

| GRANT SELECT, INSERT, UPDATE (comision, salario, oficio), DELETE ON `empresa`.`emple` TO 'adminempresa'@'localhost' |

+-----------------------------------------------------------------------------------------------------------------------+

3 rows in set (0.00 sec)

mysql> select \* from depart;

ERROR 1142 (42000): SELECT command denied to user 'adminempresa'@'localhost' for table 'depart'