

Examen 1 evaluacion acceso a datos 2019

Alice Wren y la Death Watch se han quedado encantados con tu trabajo de su web de ventas, están teniendo mucho éxito y vendiendo armas por todos los puntos de la galaxia. Cada vez más facciones quieren comprar, el Crinsom Dawn, la federación de comercio, etc.

Llegados a este punto Alice quiere tener una pequeña aplicación que le permita administrar de manera sencilla sus armas y contactos de venta. Le ofreces una herramienta muy antigua pero eficiente para este caso, una BBDD. Ella piensa que una base holográfica en 3D sería más oportuno, pero confía en tu trabajo.

Una vez te cuenta el negocio y como quiere llevarlo a cabo, se decide el modelo de datos: En términos generales, existen facciones, de las que se guarda su nombre que será la PK, la persona de contacto, el planeta donde se harán las transacciones, el número de sistemas controlados, y cuando fue su ultima compra de lo que sea. Las facciones almacenan armas para su venta a otras facciones. Las armas tienen un precio, un nombre y un id único que será la PK. Para llevar un control de las ventas, se almacenará la cantidad y fecha de las armas vendidas.

Para ampliar posibles compradores Alice quiere tener un control que todas las escaramuzas y batallas de la galaxia, almacenando las facciones participantes (si es una nueva se añadirá a BD con los datos de la faccion) el nombre de la batalla y cuando y donde se produce. Para recabar estos datos se dispone de una red de espías, distribuidas por toda la galaxia. Los espías informan de n batallas pero una batalla solo la lleva un espía. Los espías tienen nombre y raza.

JDBC:

0. (1) Cargar datos iniciales en la BD a partir de un código que se usaba en tiempos del gran creador, cuando solo existía una luz, es el XML.

1. (0,75) Siempre hay cargas familiares: el primo de Alice quedo con el cerebro dañado en una batalla; para mantenerlo ocupado, se va a dedicar a grabar armas de los rebeldes, pero el programa las borrará de inmediato.

2. (0,75) Insertar una nueva batalla, insertando la facción en caso de no existir.

```
INSERT INTO practica1.batalla (nombreBatalla, fechaBatalla, nombreEspia) VALUES ('Percula', '2020-10-01', 'Monica');
```

3. (0,75) Actualizar el precio de un arma

```
UPDATE practica1.armas t SET t.precio = 200 WHERE t.idArma LIKE '01' ESCAPE '#'
```

4. (0,5) Seleccionar todas las facciones, y todas las armas

```
SELECT * FROM Facciones, Armas
```

5. (0,5) Seleccionar el nombre y precio de las armas de una facción

```
SELECT nombreArma, precio FROM armas where nombreFaccion = 'Xore';
```

6. (0,75) Borrar los datos de un espía y sus batallas, para que si les pasa algo puedan negar su existencia.

```
DELETE FROM practica1.espia WHERE nombreEspia LIKE 'Silvia' ESCAPE '#'
```

Spring template:

7 (0,75) Seleccionar los datos de todas las armas vendidas por una facción a partir de una fecha dada.

8 (0,75) Seleccionar las batallas informadas por los espías, devolviendo nombre de batalla y nombre de espía

9 (1) Actualizar el nombre de una facción (suele pasar cuando matan al jefe y toma el control otro)

10 (0,75) Borrar aquellas armas que no se vendan

11(0,75) Borrar todos los datos de una facción.