

Sistemas de Información II – Práctica IX: Cambios en Requerimientos

Carlos Castillo / Victor Pascual – carlos.castillo@upf.edu

Modificación de requerimientos

Esta práctica es acerca de aceptar modificaciones en los requerimientos, sin necesidad de hacer todo de nuevo.

No está permitido hacer `DROP TABLE` seguido de un `CREATE TABLE`, para las tablas que ya existan, porque queremos conservar los datos que teníamos. Se trata de usar `ALTER TABLE`.

```
ALTER TABLE tabla ADD columna tipo;
ALTER TABLE tabla DROP columna;
ALTER TABLE table ADD FOREIGN KEY (columna) REFERENCES otratabla (otracolumna);
```

Obviamente, sí puede hacer `CREATE TABLE` para las tablas nuevas. Estas son las modificaciones:

- Las tarifas (euros por kilómetro) ahora están estandarizadas, es decir, son las mismas para todas las aerolíneas y tienen un nombre: Turista (0.1 euros por kilómetro), Business (0.2 euros por kilómetro) y Primera (0.4 euros por kilómetro). Recuerde actualizar los asientos en todos los vuelos para que usen estas nuevas tarifas.
- Las agencias contratan vendedores, y los vendedores están especializados, es decir, algunos vendedores sólo pueden vender pasajes de primera clase, otros sólo de Turista y Business, etc. Un mismo vendedor puede vender pasajes de varias clases. Los reservas están ahora asociadas a un vendedor, no a una agencia. Deberá crear al menos un vendedor por cada agencia y asociar los pasajes vendidos por cada agencia al primer vendedor que la agencia tenga.

Ahora viene la modificación mayor. Un analista se ha dado cuenta de que la base de datos está guardando información redundante, puesto que cada vez que se hace un vuelo se repiten las letras y los números de los asientos. Es por esto que se ha introducido la figura de “avión” que es tiene fabricante y modelo, ej.: Boeing 747. Cada avión tiene una serie de “lugares” o “plazas” que son físicamente donde se ubican los pasajeros, mientras que el concepto de “asiento” es un lugar al que se le ha asignado una tarifa, como se muestra en la Figura 1.

Un orden posible para proceder es:

- Crear la tabla lugar.
- Relacionarla con la tabla asiento.
- Copiar los datos apropiados de la tabla asiento (letra y número) usando SQL, no copiar los datos uno por uno.
- Borrar las columnas número y letra de la tabla asiento.
- Agregar la tabla avión.
- Relacionarla con la tabla lugar.

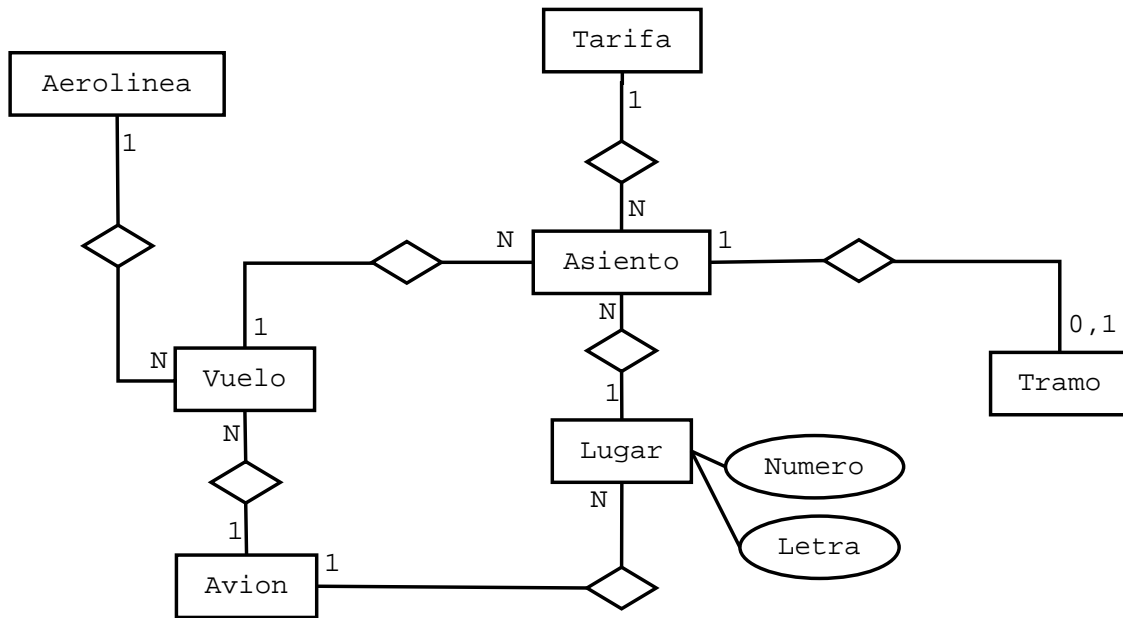


Figura 1: Efectos de incluir una entidad “avión” (vista parcial).

El resultado debe tener al menos 2 aviones con 3 lugares cada uno. Para copiar datos puede usar un INSERT con SELECT, ejemplo:

```
INSERT INTO tbldestino (col1, col2) (SELECT col3, col4 FROM tblorigen);
```

Otro fragmento de código que puede usar para copiar tablas, cuando hay identificadores únicos que copiar, utilizando la función AUTO_INCREMENT de MySQL:

```
CREATE TABLE original (id_original INT, nombre TEXT);
INSERT INTO original VALUES (100, 'X' );
INSERT INTO original VALUES (200, 'Y' );
CREATE TABLE copia (id_copia INT AUTO_INCREMENT, nombre TEXT,
PRIMARY KEY(id_copia));
INSERT INTO copia (nombre) (SELECT nombre FROM original);
SELECT * FROM copia;
```

RESULTADO

Lo que se debe entregar son:

- Todos los comandos de la práctica anterior.
- Seguidos de un comentario -- ***** -- para marcar donde comienza la parte nueva.
- Seguidos de todos los comandos de esta práctica.

Por favor verifique que copiar y pegar su fichero de comandos en el interprete SQL funciona correctamente después de ejecutar el fichero de comandos de la práctica anterior.