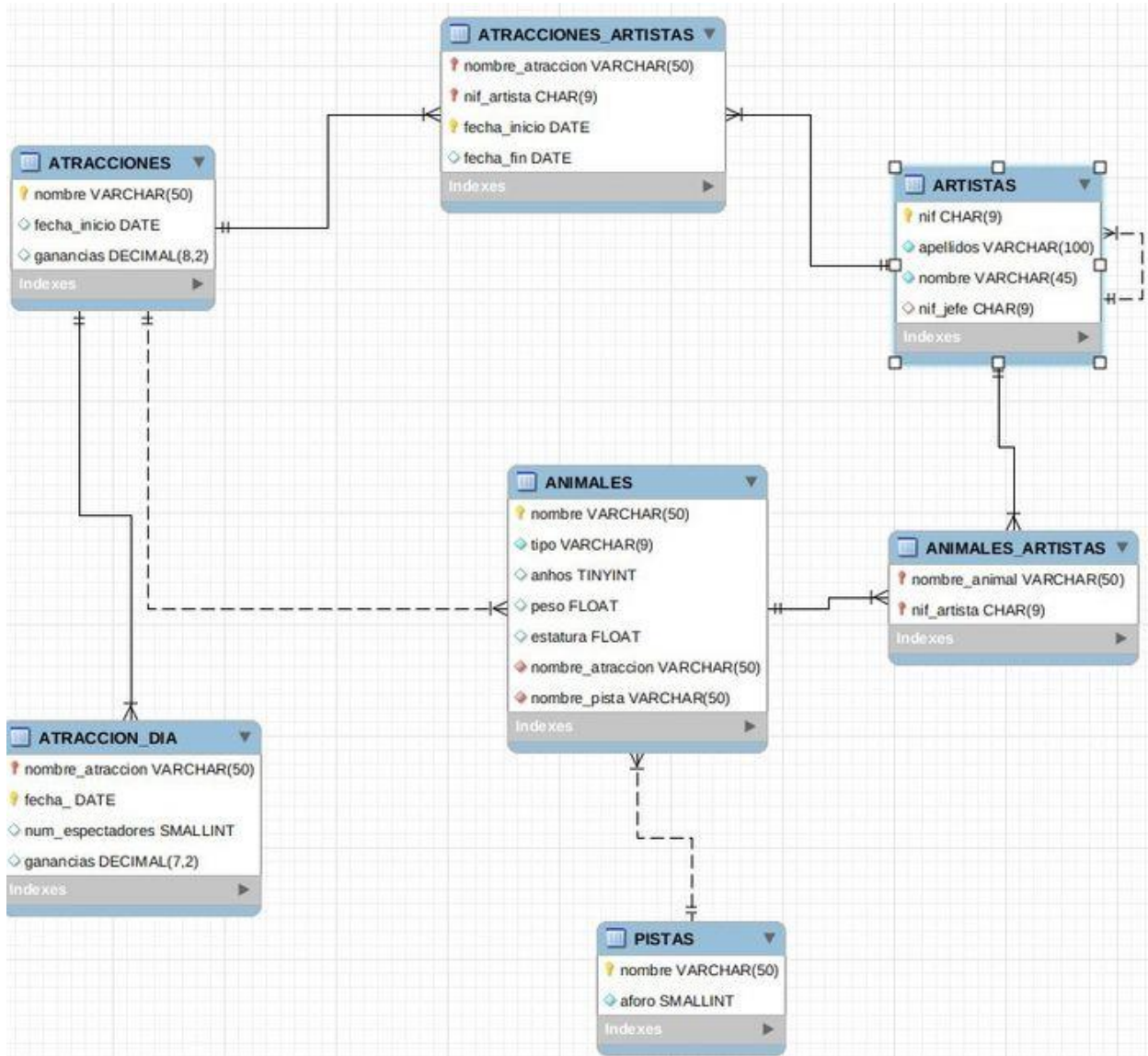


Ejercicios BD II:

Para el siguiente modelo relacional se pide lo siguiente:



Utilizar diseño de bases de datos llamado: CIRCO.sql ver archivo CIRCO.zip.

Realizar los siguientes procedimientos almacenados:

- Dado como parámetro el nif de un artista, crear un procedure que devuelva los animales que cuida dicho artista. (Valor 0.41).
- Haciendo uso de las funciones que provee Date Time, crear procedure que devuelva los datos de las atracciones que hayan comenzado hace un número de años con respecto a la fecha actual (DATE_SUB genera fechas a partir de un intervalo elegido, también deberá utilizar CURDATE e INTERVAL), para ello deberá recibir como parámetro el número de años de antigüedad. (Valor 0.41).

- Utilizando como parámetro de entrada el nombre de una pista, haga un procedure que devuelva en un parámetro de salida el aforo de dicha pista. (Valor 0.41).
- Dado el nombre de un animal, crear un procedure que devuelva un parámetro de salida con una cadena (varchar) con el formato: NombreAnimal:peso:pista:aforo (utilice la función CONCAT y el procedure anterior). (Valor 0.41).
- Dado el nombre de una pista, crear un procedure que devuelva en un parámetro de entrada/salida la cantidad que representa el incremento del aforo, es decir, el procedure debe devolver en este último parámetro el nuevo aforo de la pista. (Valor 0.41).
- Dados los parámetros de entrada/salida del nombre del animal y el incremento del aforo en la pista en la que trabaja el animal. Crear un procedure que haga uso del procedimiento anterior y devuelva empleando los dos parámetros anteriores, el nombre de la pista y su nuevo aforo. (Valor 0.45).

Realizar los siguientes Triggers:

1. Crear una tabla de nombre HISTORIAL con los siguientes atributos:

idHistoria autonumérica Clave primaria

tabla: varchar(100)

operacion: varchar(10)

data_antigua: varchar(100) (guardarán los datos nombre_pista:aforo que se borren o modifiquen)

data_nueva: varchar(100) (guardarán los datos nombre_pista:aforo que se añadan o modifiquen)

fecha-hora: datetime

Haga que se registren las operaciones de alta, baja y modificación sobre la tabla 'PISTAS'.

Ejemplos (**Nota: no utilice estos ejemplos utilice datos propios y registre hasta 10 tuplas para este caso**):

'PISTAS' - 'ALTA' - null - 'pista_nueva:1000' - '01-01-2000 17:00:00' (el campo data_antigua es nula ya que estamos a dar de alta una nueva pista).

'PISTAS' - 'BAJA' - 'pista_borrar:1000' - null - '01-01-2000 18:00:00' (el campo data_nueva es nula ya que estamos a dar de baja una pista).

'PISTAS' - 'MODIFICAR' - 'pista_modificar:1000' - 'pista_modificar:1500' - '01-01-2000 19:00:00' (el campo data_nueva guarda los datos modificados y el campo data_antigua guarda los datos antes de la modificación).

Valor 1.25

2. Crea una tabla de nombre ESTADISTICA con los siguientes atributos:

id autonumérica Clave primaria

tipo: varchar(100) no nulo

valor: int no nulo

Añade dos filas con los valores para tipo/valor: pistas/0 animales/0

Así:

```
INSERT INTO `CIRCO`.`ESTADISTICA` (`tipo`, `valor`) VALUES ('pistas', '0');
```

```
INSERT INTO `CIRCO`.`ESTADISTICA` (`tipo`, `valor`) VALUES ('animales', '0');
```

Haga un disparador para que cada vez que haya alguna operación que modifique (alta/baja) el número de pistas o de animales, se actualice el número total de los mismos.

Ejecutar las ordenes SQL que actualicen la tabla ESTADISTICA con los datos actuales de las tablas (**Nota: realice 10 operaciones con datos propios de modificación para activar el disparador**).

Valor 1.25