



**Examen de Bases de datos**  
**Grado de Ingeniería en Informática**  
**3 de septiembre, 2015**

Nombre:

1) (3 puntos) Un parque de atracciones desea implementar una base de datos relacional que cumpla los siguientes requisitos:

El parque de atracciones abre una cantidad de días al año que debe conocerse de antemano, así como la hora de apertura y cierre, y el número de entradas ya vendidas para dicho día.

Debe tener una lista de atracciones ofertadas con un nombre único, un aforo máximo, un tipo (maquinismo, infantil, etc.), el número de encargados necesarios para su funcionamiento y un número de ubicación de la atracción en un mapa que se ofrece a los visitantes.

Debe contratar a los encargados, conociendo su DNI, nombre, apellido, fecha de nacimiento y fecha de contrato.

Cada día de apertura se debe conocer que encargados se ocupan de cada atracción y cuantas horas está cada encargado en cada atracción.

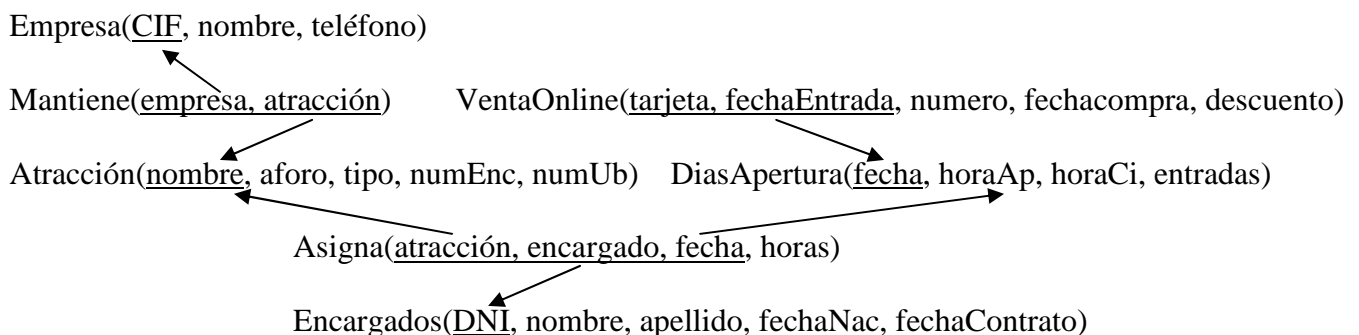
En el mismo día de apertura, un encargado no podrá tener asignados más de tres atracciones. Los encargados son eventuales y eligen los días de apertura que quieren trabajar (no necesariamente todos).

Se contrata el mantenimiento de las atracciones a varias empresas, cada una de las cuales se especializa en mantener algunas. De cada empresa se debe saber el CIF, el nombre y un teléfono de contacto. Cada atracción debe tener asignada al menos una empresa de mantenimiento.

Se hacen ventas online de entradas que incluyen la tarjeta del cliente (necesaria para la recogida), número de entradas, la fecha de compra, la fecha de la entrada y un descuento variable para cada venta (que se calcula según el número de entradas ya vendidas para dicho día). Con una tarjeta sólo se puede comprar entradas para un día como mucho una vez, aunque se puede modificar el número de entradas de cada día.

- (1,5 puntos) Se pide diseñar el esquema conceptual con el modelo entidad-relación especificando las restricciones de participación mínimo-máximo o cardinalidad.
- (1 puntos) Traducir dicho esquema a relacional e incluir las reglas de integridad referencial.

Solución:



- (0,5 puntos) Indicar las restricciones no reflejadas en el esquema relacional.

Solución:

En el mismo día de apertura, un encargado no podrá tener asignados más de tres atracciones.  
Cada atracción debe tener asignada al menos una empresa de mantenimiento.

Un encargado en un día puede tener asignado hasta 3 atracciones



Una atracción tiene al menos una empresa asignada

2) (2,5 puntos) Dadas las relaciones (no necesariamente del esquema anterior):

Encargado( DNI, nombre, apellido, fechaDeNacimiento, fechaDeContrato)  
Trabaja( DNI, atracción, fecha, horas)

Escribir las siguientes consultas:

$\Pi_{\text{nombre, apellido}}(\text{Encargado} \bowtie (\sigma_{\text{atracción}='montaña rusa' \text{ AND } \text{mes}=1 \text{ AND } \text{año}=2015}(\text{Puertos})))$

- (0,5 puntos) SQL: Mostrar sin repetir el nombre y apellido de todos los encargados que trabajan en 'la montaña rusa' en enero de 2015.

```
SELECT DISTINCT nombre, apellido
FROM Encargado E, Trabaja T
WHERE E.DNI=T.DNI
      AND atracción='montaña rusa'
      AND MONTH(fecha)=1
      AND YEAR(fecha)=2015
```

- (0,5 puntos) SQL: Mostrar en una sola consulta el nombre y apellido del empleado más mayor, el que lleva más tiempo trabajando y el que en total lleva más horas asignadas. Indicar en la salida el tipo de premio que se les concederá a cada uno: 'veterano', 'antigüedad' o 'másHoras'.

```
SELECT nombre, apellido, 'veterano' AS TipoPremio
FROM Encargado
WHERE fechaDeNacimiento= (SELECT MIN(fechaDeNacimiento)
                           FROM Encargado
                           )

UNION ALL

SELECT nombre, apellido, 'antigüedad' AS TipoPremio
FROM Encargado
WHERE fechaDeContrato= (SELECT MIN(fechaDeContrato)
                        FROM Encargado
                        )

UNION ALL

SELECT nombre, apellido, 'másHoras' AS TipoPremio
FROM Encargado E, Trabaja T
WHERE E.DNI=T.DNI
GROUP BY nombre, apellido
HAVING SUM(horas)>= ALL (SELECT SUM(horas)
                        FROM Encargado E, Trabaja T
                        WHERE E.DNI=T.DNI
                        GROUP BY nombre, apellido
                        )
```

- (0,5 puntos) SQL: Mostrar en el nombre y apellido de todos los encargados con el número de atracciones distintas y número de horas totales asignadas en enero de 2015 (que puede ser cero).

```
SELECT nombre, apellido, COUNT(DISTINCT atracción)AS atracciones,
      SUM (horas) AS Horas
```



```
FROM Encargado E, Trabaja T
WHERE E.DNI=T.DNI
GROUP BY nombre, apellido
UNION ALL
SELECT nombre, apellido, 0, 0
FROM Encargado E
WHERE NOT EXISTS (SELECT *
                  FROM Trabaja
                  WHERE DNI = E.DNI)
```

- (0,5 puntos) SQL: Mostrar para cada día de apertura de enero de 2015 las atracciones (ya incluidas en Trabaja) con el número de encargados asignados y la suma de horas asignadas. La salida debe estar ordenada por atracción y día.

```
SELECT atracción, fecha, COUNT(DISTINCT DNI)AS encargados,
        SUM (horas) AS Horas
FROM Trabaja
WHERE MONTH(fecha)= 1
AND año=2015
GROUP BY atracción, fecha
ORDER BY 1,2
```

- (0,5 puntos) SQL: Borrar los encargados que no han tenido ninguna asignación en 2014.

```
DELETE FROM Encargados
WHERE NOT EXISTS (SELECT *
                 FROM Trabaja
                 WHERE DNI = E.DNI)
```

- 3) (2,5 puntos) Dado el esquema (no necesariamente de ejercicios anteriores):

Trabaja( DNI, atracción, fecha, horas)  
DiasApertura( fecha, horaApertura, horaCierre, entradasVendidas)  
VentasOnline(tarjeta, fecha, numEntradas, fechaCompra, descuento)

- (0.5 puntos) Crear las tablas con SQL incluyendo integridad referencial y las siguientes restricciones
  - Si se decide cerrar un día por mal tiempo se borran en cascada las asignaciones de trabajo y VentasOnline de dicho día (los 'delete triggers' se ocupan de informar a los encargados y devolver el dinero a los visitantes).
  - No se puede asignar una atracción a un encargado más de 8 horas al día.

### Solución

```
Create table Trabaja
(DNI CHAR (9),
atraccion char(25),
fecha DATE,
horas NUMBER(1),
CONSTRAINT tra_PK PRIMARY KEY (dni, atracción, fecha),
CONSTRAINT tra_ct_FK FOREIGN KEY (fecha) REFERENCES
DiasApertura on delete cascade,
CONSTRAINT tra_maximo CHECK(horas<=8))
```



```
Create table VentasOnline
  (tarjeta CHAR (10),
  fecha DATE,
  fechaCompra DATE,
  numEntradas NUMBER(4),
  Descuento NUMBER (10,5),
  CONSTRAINT tra_PK PRIMARY KEY (tarjeta, fecha),
  CONSTRAINT tra_ct_FK FOREIGN KEY (fecha) REFERENCES
  DiasApertura on delete cascade)
```

```
Create table VentasOnline
  (fecha DATE,
  HoraApertura TIME,
  HoraCierre TIME,
  EntradasVendidas NUMBER(5)
  CONSTRAINT tra_PK PRIMARY KEY (fecha))
```

- (1 punto) Escribir un procedimiento almacenado que reciba por argumento una fecha de apertura y muestre por consola las atracciones que no tienen asignadas a encargados suficientes horas para cubrir completamente el horario de apertura de dicho día.

```
CREATE OR REPLACE PROCEDURE Asignadas ( p_día DATE) IS
CURSOR horasAtraccion IS
  SELECT atraccion, SUM(horas) AS Horas
  FROM Trabaja
  WHERE trabaja.fecha=p_dia
  GROUP BY atracción;

BEGIN

  FOR total IN horasAtraccion LOOP
    IF total.Horas < (SELECT horaCierre-HoraApertura AS Horario
                     FROM DiasApertura,
                     Where DiasApertura.fecha = p_dia)
    THEN DBMS_OUTPUT.PUT_LINE( 'No hay suficientes encargados para
  ` || total.atraccion);
    END IF;
  END LOOP;
END;
```

- (1 punto) Crear un disparador (trigger) que mantenga actualizado el atributo 'entradasVendidas' en DiasApertura al insertar, modificar el número de entradas, o borrar ventas online.

```
CREATE OR REPLACE TRIGGER ActualizaDiasApertura
AFTER INSERT OR UPDATE OR DELETE ON VentasOnline
FOR EACH ROW
BEGIN
  IF INSERTING
  UPDATE DiasApertura
  SET EntradasVendidas= EntradasVendidas + :new.numEntradas
  WHERE día=:new.dia
END IF;
END;
```



4) (1 punto) Describir qué es un Deadlock (autobloqueo) y cómo lo resuelve el gestor de bloqueos.

5) (1 puntos) Dado el esquema (no necesariamente el de ejercicios anteriores):

Trabaja( DNI, atracción, fecha, horas, aforoAtracción)

Descomponer Trabaja hasta llegar a 3FN subrayando las claves.

Solución:

Trabaja( DNI, atracción, fecha, horas)

Atracción(atracción, aforoAtracción)