Ejercicios Parciales Fundamentos de Bases de Datos

2023

SOLUCIÓN

Ejercicio 1

Se desea modelar una base de datos con información sobre parques de diversiones.

Cada parque de diversión tiene un código que lo identifica, un nombre, capacidad máxima de visitantes, horarios de apertura (Ejemplo: viernes 8:00-18:00, sábado 10:00-22:00, etc.) y la ubicación (calle, número de puerta y ciudad). Los parques de diversión contienen distintas atracciones, las cuales poseen un código que las identifica dentro del parque. Además, tienen un nombre, tipo (montaña rusa, infantil, etc.) y altura mínima requerida para ingresar.

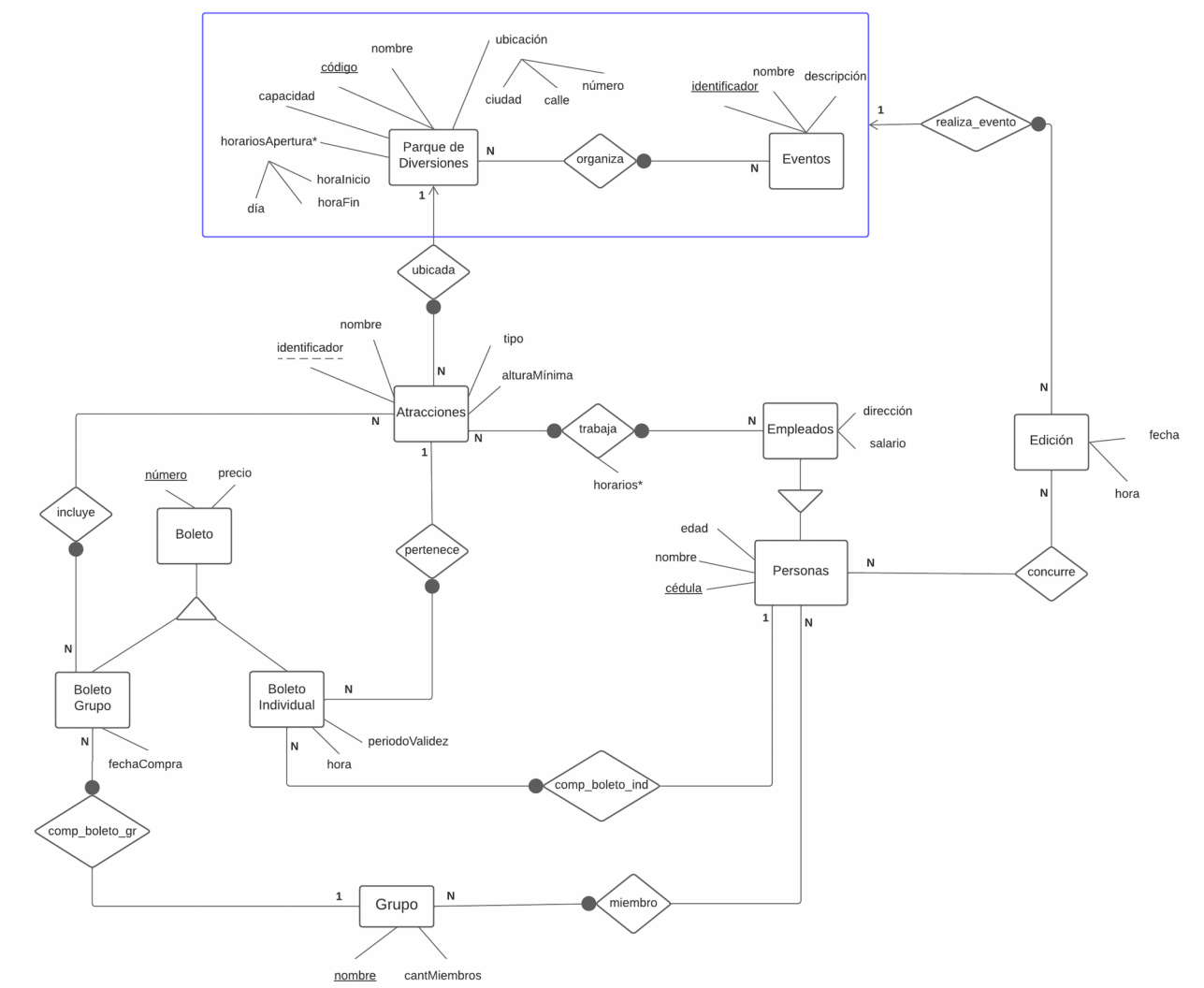
En las atracciones trabajan personas (empleados) y para cada una de ellas se quiere almacenar los horarios de trabajo en cada atracción (pudiendo una persona trabajar en varias atracciones). De las personas se conoce la cédula, el nombre y la edad. Además, de los empleados se conoce su dirección y salario. Existen grupos de personas, los cuales se identifican por su nombre y también interesa almacenar el total de miembros y las personas que integran cada grupo.

La entrada a los parques es gratuita, pero para acceder a una atracción es necesario comprar un boleto. Los boletos tienen un número que los identifica y un precio. Los boletos individuales los compra una persona y valen para una atracción, y tienen una hora y período de validez (Ejemplo: 15hs. – 2 horas). Estos boletos solo los pueden comprar personas mayores de 18 años. Por otro lado, también existen boletos grupales, que son comprados por los grupos, los cuales son válidos para un conjunto de atracciones dentro de un mismo parque. De estos boletos se conoce la fecha de compra.

Por último, los parques organizan distintos eventos (Ejemplo: conciertos, festivales, desfiles, etc.) de los cuales se conoce un identificador, el nombre y la descripción. Un mismo evento se puede repetir en el tiempo y realizarse en varios parques (Ejemplo: El desfile de Navidad se hace todos los años en el Parque 1, Parque 2 y Parque 3), siendo cada una de esas una edición diferente. De cada edición del evento, se quiere almacenar la fecha y hora en la cual se realiza. Se quiere registrar a qué edición del evento concurren las personas.

Se pide: Modelo Entidad-Relación completo del problema.

Solución:


Parque de Diversiones
nombre
código
ubicación
ciudad
calle
número
capacidad
horariosApertura*
horaInicio
horaFin
día
organiza
Eventos
identificador
nombre
descripción
realiza_evento
Edición
fecha
hora
ubicada
Atracciones
nombre
identificador
tipo
alturaMínima
trabaja
horarios*
Empleados
dirección
salario
incluye
Boleto
número
precio
pertenece
Personas
edad
nombre
cédula
concurre
Boleto Grupo
Boleto Individual
fechaCompra
periodoValidez
hora
comp_boleto_ind
comp_boleto_gr
Grupo
nombre
cantMiembros
miembro

RNEs:

1. Las atracciones asociadas a un Boleto de Grupo deben pertenecer al mismo Parque de Diversiones:

   $(\forall bg \in BoletoGrupo)(\forall a_1 \in Atracción)(\langle bg, a_1\rangle \in incluye \wedge$
   $(\exists a_2 \in Atracción)(\langle bg, a_2\rangle \in incluye) \rightarrow ubicada(a_1) = ubicada(a_2))$

2. Las personas que compran boletos individuales son mayores de edad (mayores de 18 años)

   $(\forall p \in Persona)(\forall bi \in BoletoIndividual)(\langle p, bi\rangle \in compra\_boleto\_individual \rightarrow edad(p) \geq 18)$

3. El atributo *cantidadMiembros* se calcula como la suma de las instancias de Persona asociadas al Grupo mediante la relación **miembro**

   $$(\forall g \in Grupo)(\sum_{p \in Persona : \langle p,g\rangle \in miembro} 1 = cantMiembros(g))$$

4. Una persona no pueden estar en 2 ediciones distintas que sucedan al mismo tiempo.

   $(\forall p \in Persona)(\forall e_1, e_2 \in Edicion)((< p, e_1 > \in concurre \wedge$
   $< p, e_2 > \in concurre \wedge fecha(e_1) = fecha(e_2) \wedge hora(e_1) = hora(e_2)) \rightarrow$
   $realiza\_evento(e_1) = realiza\_evento(e_2))$

5. $Boleto = BoletoIndividual \cup BoletoGrupo$

6. $BoletoIndividual \cap BoletoGrupo = \emptyset$

Ejercicio 2

Considere el siguiente esquema relacional de una agencia de viajes.

CLIENTES (ciCliente, nomCliente, telefono, direccion, edad)
Datos personales de los clientes.

PASAJES (codPasaje, ciCliente, destino, aerolinea, fechaSalida, fechaRegreso)
Datos de cada pasaje vendido a un cliente.

HOTELES (codHotel, nomHotel, ciudad, pais, web)
Datos de los hoteles con los que trabaja la agencia.

PAQUETES (codPaquete, codHotel, duracion, precio)
Datos de los paquetes que ofrece la agencia cuando se compra un pasaje.

PASAJE-PAQ (codPasaje, codPaquete, fecha-vta)
Datos de los paquetes que se vendieron a los clientes junto a los pasajes. Corresponde a los casos en que el cliente compró un paquete asociado al pasaje.

$$\Pi_{ciCliente}(PASAJES) \subseteq \Pi_{ciCliente}(CLIENTES)$$
$$\Pi_{codHotel}(PAQUETES) \subseteq \Pi_{codHotel}(HOTELES)$$
$$\Pi_{codPaquete}(PASAJE-PAQ) \subseteq \Pi_{codPaquete}(PAQUETES)$$
$$\Pi_{codPasaje}(PASAJE-PAQ) \subseteq \Pi_{codPasaje}(PASAJES)$$

Nota: Ninguna de las tablas se encuentra vacía.

Se pide:

1. **Resolver en Álgebra Relacional las siguientes consultas**

   a) Devolver los nombres de los hoteles que solamente se encuentran en paquetes cuya duracion es mayor a 10 días.

   A = $\Pi_{codHotel}$ ($\sigma_{duracion \leq 10}$ (Paquetes))

   SOL = $\Pi_{nomHotel}$ (($\Pi_{codHotel}$ (Paquetes) – A) * Hoteles)

   b) Devolver los códigos de los paquetes que fueron vendidos antes del 13/3/2020 al precio más bajo de los vendidos en ese período.

   A = $\Pi_{codPaquete,\ precio}$ ($\sigma_{fecha\text{-}vta \leq 13/3/2020}$ (Pasaje-Paq) * Paquetes)

   B = $\rho_{\ codPaquete,\ precio \rightarrow CP,\ p}$ (A)

   C = Π_{CP} ($\sigma_{p > precio}$ (B x A))

   SOL = $\Pi_{codPaquete}$ (A) – C

   Otra solución

   A = $\Pi_{codPaquete,\ precio}$ ($\sigma_{fecha\text{-}vta \leq 13/3/2020}$ (Pasaje-Paq) * Paquetes)

$B = \rho_{\text{codPaquete, precio} \rightarrow \text{CP, p}}(A)$

$C = \Pi_{\text{codPaquete,CP}}(\sigma_{\text{precio}<=p}(A \times B))$

$SOL = C \,\%\, \Pi_{\text{codPaquete}}(A)$

2. Resolver en SQL las siguientes consultas

a) Devolver los códigos de paquetes que fueron vendidos a más de 30 personas menores de 18 años.

```
SELECT codPaquete
FROM Pasaje-Paq P1 NATURAL JOIN Pasajes P2 NATURAL JOIN Clientes C
WHERE C.edad < 18
GROUP BY codPaquete
HAVING COUNT(distinct ciCliente) > 30
```

b) Devolver los nombres de clientes que compraron estadías en todos los hoteles de Viena ofrecidos por la agencia.

```
SELECT nomCliente
FROM Clientes C
WHERE NOT EXISTS
        (SELECT *
        FROM Hoteles H
        WHERE ciudad = 'Viena' AND
            NOT EXISTS
                (SELECT *
                FROM Pasaje-Paq P1 NATURAL JOIN Pasajes P2
                        NATURAL JOIN Paquetes P3
                WHERE P2.ciCliente = C.ciCliente AND P3.codHotel = H.codHotel))
```

Otra solución:

```
SELECT nomCliente
FROM Clientes C
WHERE NOT EXISTS
        (SELECT *
        FROM Hoteles H
        WHERE H.ciudad = 'Viena' AND
            H.codHotel NOT IN
                        (SELECT P3.codHotel
                        FROM Pasaje-Paq P1 NATURAL JOIN Pasajes P2
                                NATURAL JOIN Paquetes P3
                        WHERE P2.ciCliente = C.ciCliente))
```