

Ejercicios Parciales de Fundamentos de Bases de Datos

2022

Ejercicio 1. (20 puntos)

Para este ejercicio, considere la siguiente descripción de la realidad relativa a un sitio web que se dedica a alquilar vehículos.

Cuando las personas se registran en el sitio, deben llenar un formulario que tiene la cédula de identidad, su nombre y otros datos. A su vez las personas pueden ser arrendadores o dueñas de vehículos. Cualquier persona puede ser un arrendador pero para ser dueño es necesario tener al menos un vehículo registrado como de su propiedad. Cada persona puede registrarse como propietaria de varios vehículos y cada vehículo puede tener varios dueños.

Los vehículos está identificados por un código que le asigna el sitio web. Además se conocen otros datos del vehículo. Todos los vehículos tienen al menos un dueño.

Cuando un arrendador alquila un vehículo, se genera un contrato que incluye al arrendador, el vehículo y solamente a uno de los dueños de ese vehículo. Además se conoce en qué día se inicia el contrato, qué día finaliza y el precio acordado. Para cada contrato, también se conoce un número. Ese número no puede repetirse para cada dueño y vehículo, pero, sí puede estar repetido para el mismo dueño con otros vehículos o incluso dueños diferentes con vehículos diferentes. La fecha de inicio de cualquier contrato debe ser menor que su fecha de fin. Por último, se sabe que pueden existir vehículos y dueños registrados que no hayan participado en ningún contrato de alquiler. Tenga en cuenta que ningún dueño puede arrendar un vehículo de su propiedad.

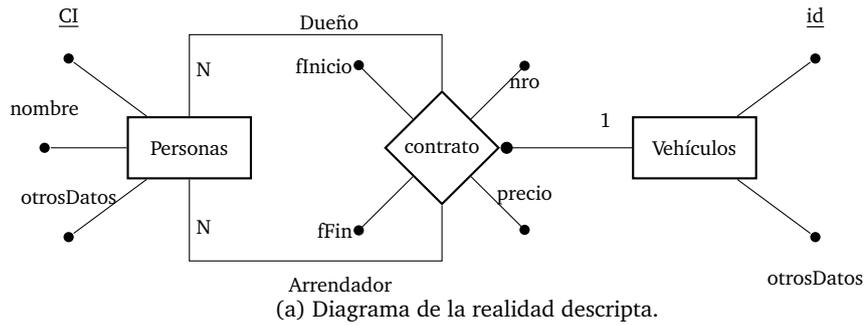
- (a) Considere el diagrama de la Figura 1a y la restricción no estructural de la Figura 1b. Expresé en lenguaje natural dicha restricción.
- (b) Indique si las siguientes afirmaciones sobre el diagrama de la Figura 1a son verdaderas o falsas. Justifique su respuesta.
 - I. La relación *contrato* no permite que existan parejas de *Vehículos* y *Dueños* sin que participe un *Arrendador*.
 - II. La fecha de inicio de los contratos es menor o igual que la fecha de fin.
 - III. El número de contrato puede repetirse para un dueño y un vehículo con distinto arrendador.
 - IV. Un dueño puede arrendar un vehículo de su propiedad.
- (c) Considere ahora algunas precisiones y agregados a la realidad descrita anteriormente. Los otros datos de las personas son sus teléfonos y su dirección que está formada por el país, la ciudad, y la calle y número. Además, de los arrendadores se conoce el tipo de libreta de conducir (nacional, internacional). Los otros datos que se conocen de los vehículos son su marca, modelo, año de la última revisión y matrícula. Tenga en cuenta que no hay matrículas que se repitan entre distintos vehículos registrados. Además, los vehículos pueden ser solo Familiares o Utilitarios. De los Familiares se conoce la cantidad de pasajeros que pueden viajar y de los Utilitarios, la cantidad de toneladas que pueden cargar. El sitio no permite registrar vehículos Utilitarios como Familiares ni Familiares como Utilitarios.

Se pide: Construir un esquema Entidad-Relación completo incluyendo las Restricciones No Estructurales. Incluya la restricción de la parte a adaptada a su diagrama.

Ejercicio 2. (10 puntos)

Considere el siguiente esquema relacional, similar al utilizado en la Tarea 2 del laboratorio

MOVIES (id, name, date, runtime, budget, revenue, homepage, vote_average)



$$\begin{aligned}
 & (\forall v \in Vehículos)(\forall d_1 \in Personas)(\forall a_1 \in Personas) \\
 & (\forall d_2 \in Personas)(\forall a_2 \in Personas)(\\
 & \quad \langle d_1, a_1, v \rangle \in contrato \wedge \langle d_2, a_2, v \rangle \in contrato \wedge (d_1 \neq d_2 \vee a_1 \neq a_2) \rightarrow \\
 & \quad (fFin(\langle d_1, a_1, v \rangle) < fIni(\langle d_2, a_2, v \rangle) \vee \\
 & \quad fFin(\langle d_2, a_2, v \rangle) < fIni(\langle d_1, a_1, v \rangle)) \\
 &)
 \end{aligned}$$

(b) Restricción no Estructural.

Figura 1: Diagrama y Restricción no Estructural a considerar en la Parte a.

Contiene información sobre películas.

SAGAS (id, name, date)

Contiene información sobre las sagas de películas.

PEOPLE (id, name, birthdate, deathdate, gender)

Contiene información sobre las personas que trabajan en películas.

GENRES (id, name)

Contiene información sobre los géneros de películas.

JOBS (id, name)

Contiene información sobre las tareas que se realizan en las películas.

CASTS (movie_id, person_id, job_id, role)

Contiene, para cada película, registros sobre las personas que trabajan en ellas, las tareas que realizan y el rol que desempeñan.

MOVIE_GENRES (movie_id, genre_id)

Indica los géneros de las películas.

MOVIE_IN_SAGA (movie_id, saga_id)

Indica la pertenencia de las películas a las sagas.

En este esquema se cumplen las siguientes dependencias de inclusión:

$$\begin{aligned}
\Pi_{movie_id}(CASTS) &\subseteq \Pi_{id}(MOVIES) \\
\Pi_{person_id}(CASTS) &\subseteq \Pi_{id}(PEOPLE) \\
\Pi_{job_id}(CASTS) &\subseteq \Pi_{id}(JOBS) \\
\Pi_{movie_id}(MOVIE_GENRES) &\subseteq \Pi_{id}(MOVIES) \\
\Pi_{genre_id}(MOVIE_GENRES) &\subseteq \Pi_{id}(GENRES) \\
\Pi_{movie_id}(MOVIE_IN_SAGA) &\subseteq \Pi_{id}(MOVIES) \\
\Pi_{saga_id}(MOVIE_IN_SAGA) &\subseteq \Pi_{id}(SAGAS)
\end{aligned}$$

- (a) Resolver en Álgebra Relacional la siguiente consulta:
- I. Devolver identificador de persona, identificador de tarea (job) y nombre de las tareas realizadas por personas en películas de aventuras tales que esas personas trabajaron en películas de aventuras y no trabajaron en películas de acción. NOTA: el identificador del género aventura es 12 y el de acción es 28.
- (b) Resolver en SQL sin utilizar vistas, ni subconsultas en el FROM ni la expresión WITH la siguiente consulta:
- I. Nombre de las sagas y cantidad de películas que la componen para aquellas sagas en las que trabajaron más de 30 personas distintas.
- (c) Dar una expresión equivalente en Álgebra relacional a la siguiente consulta en SQL:

```

I.      select m.name , m.budget
        from movies m
        where m.date >= '1/1/2022'
        and not exists
          (select 1
           from jobs j
           where not exists
             (select 1
              from casts c
              where c.movie_id = m.id
                and c.job_id = j.id
              )
          )

```