Ayudantía Interrogación 1

Antonio Ossa @ IIC2413 Bases de Datos

Pregunta 1

Modelo relacional

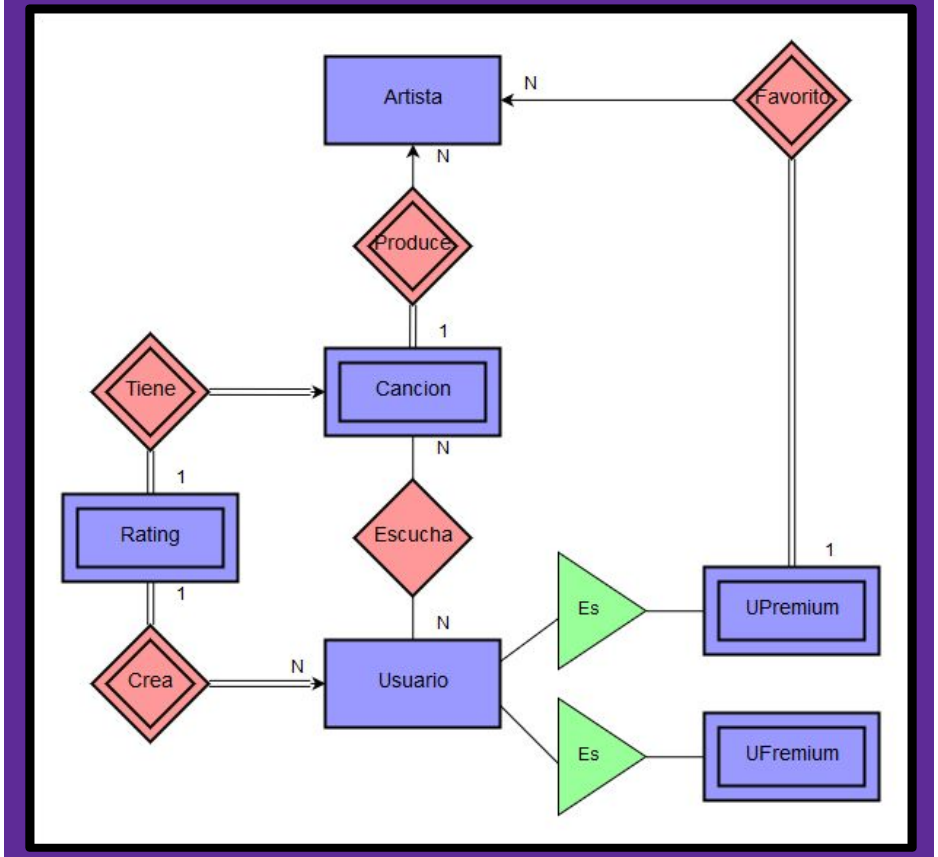
A partir del texto, construir el diagrama E/R, con los atributos que se estimen apropiados.

En un servicio de *streaming* de música, se almacena cada vez que un usuario escucha una canción (a la que además, pueden colocar una nota). La colección de canciones (cada una posee su duración, nombre, año de lanzamiento y género) está organizada por artista (cada uno con su foto y nombre). Los usuarios pueden tener un perfil “freemium” (solo sabemos cuantos comerciales han escuchado) o “premium” (que tienen su foto de perfil y un artista favorito visible).

Pregunta 1

Modelo relacional

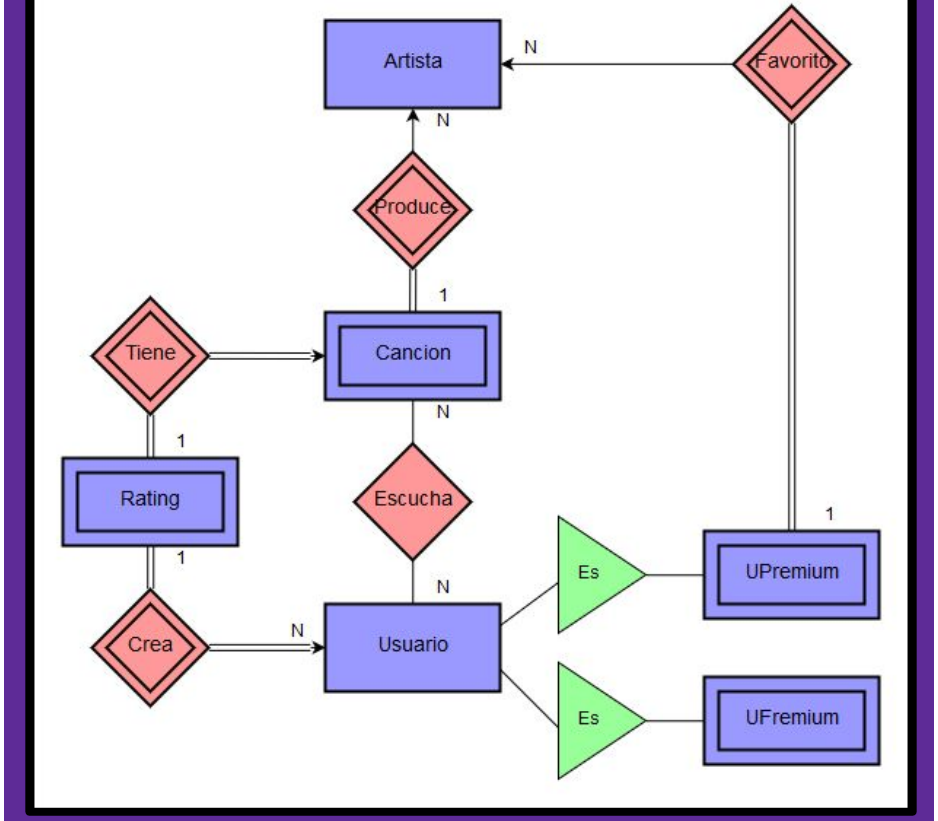
A partir del texto, construir el diagrama E/R, con los atributos que se estimen apropiados.



Pregunta 1

Modelo relacional

A partir del diagrama, construir el esquema SQL apropiado.



Pregunta 1

Modelo relacional

A partir del diagrama, construir el esquema SQL apropiado.

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 2

Álgebra relacional

Escribir la consulta que entregue “el nombre de cada canción junto al nombre de su artista”, en álgebra relacional

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 2

Álgebra relacional

Escribir la consulta que entregue “el nombre de todas las personas que han escuchado canciones de ‘Daft Punk’”, en álgebra relacional

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 2

Álgebra relacional

Se define el operador $Similar_k(Escucha)$ que funciona retornando todos los pares de usuarios que escuharon exactamente las mismas *k* canciones.
¿Es monónoto?

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 2

Álgebra relacional

Ahora tenemos el operador $SimilarDesde_k(Escucha)$ que funciona retornando todos los pares de usuarios que escuharon al menos las mismas k canciones.
¿Es monónoto?

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 2

Álgebra relacional

De forma general, ¿cómo demostraría que un operador es monónoto? ¿Y si no fuera monótono?

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 3

Uso de SQL

Escribir en SQL la siguiente consulta:
“Canciones que se llamen igual pero que sean de distintos artistas”

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 3

Uso de SQL

Escribir en SQL la siguiente consulta:
“Promedio de *ratings* para la canción Y.M.C.A de Village People”

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 3

Uso de SQL

Escribir en SQL la siguiente consulta:
“Los nombres de los usuarios cuyo promedio de *ratings* es mayor que 5 y han entregado al menos 3 *ratings*”

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)

Pregunta 3

Uso de SQL

Escribir en SQL la siguiente consulta:
“La tercera canción con mejor promedio de *ratings* (sin usar *LIMIT* y pensando en extraer el k-ésimo valor más alto)”

- Cancion(cancion_id, artista_id, nombre_cancion, duracion)
- Artista(artista_id, nombre_artista, foto)
- Usuario(usuario_id, email, nombre_usuario)
- UPremium(usuario_id, foto, artista_id)
- UFreemium(usuario_id, n_ads_vistos)
- Escucha(cancion_id, usuario_id, fecha)
- Rating(usuario_id, cancion_id, nota)