Tarea 2 - Consultas

Bases de Datos para Ingeniería - Laboratorio 2024

Objetivos: Realizar consultas en SQL, interactuando con una base de datos alojada en un servidor de base de datos PostgreSQL.

Material de referencia:

1. Material del curso.
2. Capítulos 6 y 7, Fundamentals of Database Systems, Elmasri & Navathe, 7th Edition, Pearson, 2016
3. Tutorial PostgreSQL

Formato de entrega:

1. Las consultas 1-9 se deben entregar en un archivo con extensión .sql que indique el número de la consulta (Por ejemplo, la solución para la Consulta 1 se almacena en un archivo denominado consulta1.sql).
   Cada uno de estos archivos sólo debe contener **una consulta y el esquema debe coincidir con el indicado en la letra (mismo nombre y orden de columnas)**.
2. La consulta 10 se debe entregar en un archivo PDF.
   IMPORTANTE: En este PDF, indicar el número de grupo y los integrantes

Entrega: Cada grupo podrá entregar su solución hasta el **7/11/2024 a las 23:59 hs**.

1. Introducción

En la Sección 2 se presenta una descripción de la realidad sobre la que se trabajará, mientas que en la Sección 3 se describe el esquema relacional a utilizar y se brindan instrucciones para su creación y carga. Por último, en la Sección 4 se detallan algunas restricciones a tener en cuenta para la resolución de las consultas, mientras que la Sección 5 plantea las consultas a resolver.

2. Descripción de la Realidad

En esta tarea trabajaremos sobre una base de datos que almacena información de una tienda de contenidos multimedia (Ver Figura 1), incluyendo datos de artistas (*artist*), álbumes (*album*), canciones (*track*), listas de reproducción (*playlist*) donde se encuentran las canciones, empleados (*employee*), clientes (*customer*) y facturas (*invoice*).

Se tiene información de los empleados de la tienda, de los cuales se conoce un identificador, un nombre, su información de residencia, puesto de trabajo y a quién reporta.

También se conoce información de los clientes de la tienda, como el identificador, el nombre y su información de residencia. A su vez, la tienda lleva un registro de las compras realizadas por cada cliente.

La tienda permite comprar canciones de las cuales se conoce el álbum al que pertenece, el género (*genre*), el tipo de medio (*media type*) (por ejemplo, MP3, AAC, entre otros.) y las listas de reproducción a las que pertenecen.

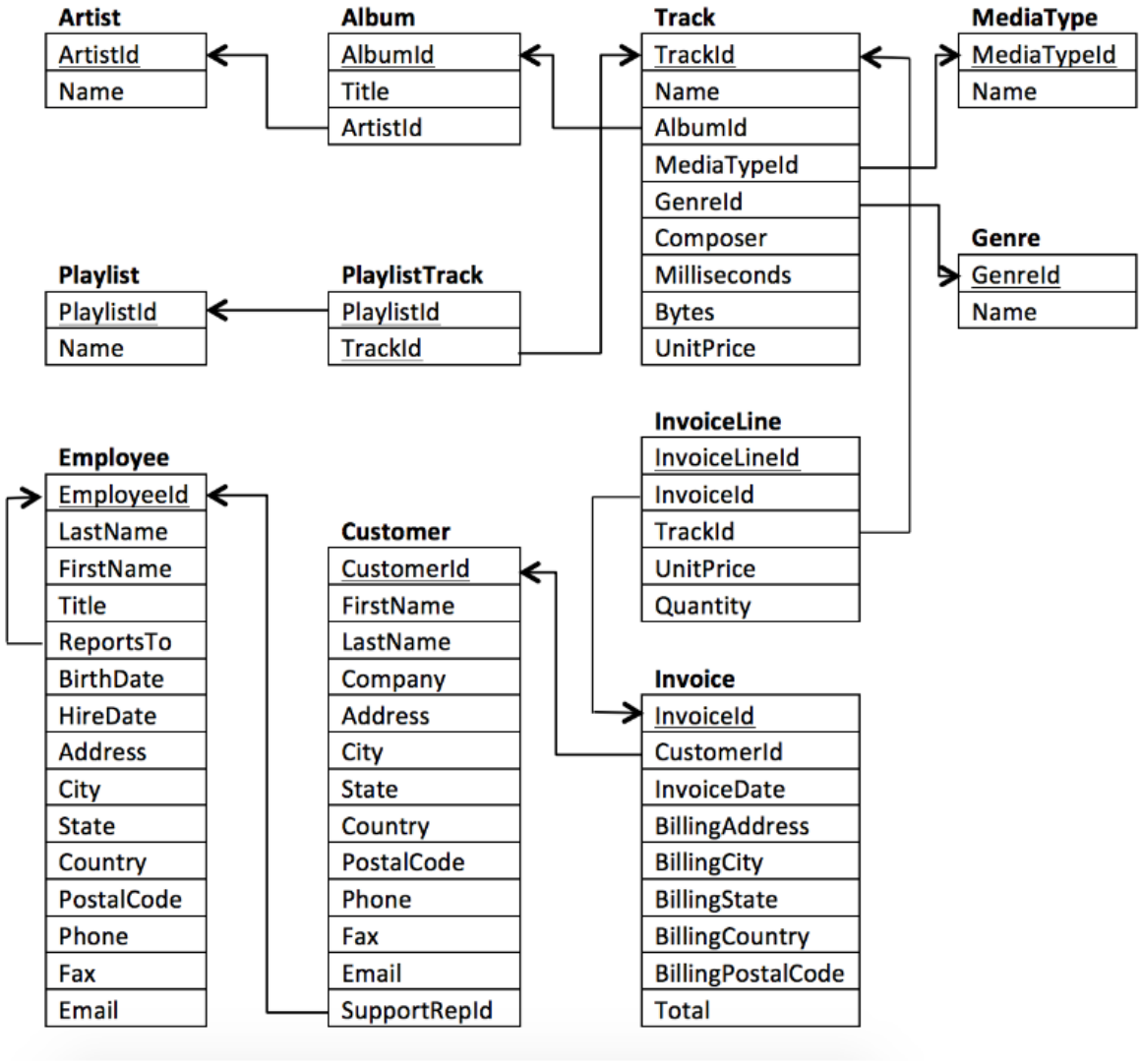

Figura 1: Modelo relacional de la base de datos Chinook

3. Esquema relacional y datos

A continuación se presenta el esquema relacional sobre el cual trabajaremos.

Esquema Relacional

- **employee (employee_id, last_name, first_name, title, reports_to, birth_date, hire_date, address, city, state, country, postal_code, phone, fax, email)**
  Contiene información sobre los empleados: su identificador, su nombre y apellido, su puesto de trabajo,

a quién reporta, su fecha de nacimiento, su fecha de contratación, información de residencia (dirección, estado, país y código postal), teléfono celular, número de fax y correo electrónico.

- **customer (customer_id, first_name, last_name, company, address, city, state, country, postal_code, phone, fax, email, support_rep_id)**
  Almacena información sobre los clientes de la tienda: su identificador, su nombre y apellido, la compañía en la que trabaja, información de residencia (dirección, ciudad, estado, país y código postal), teléfono celular, número de fax, correo electrónico y el empleado encargado de asesorarlo en la tienda.
- **invoice (invoice_id, customer_id, invoice_date, billing_address, billing_city, billing_state, billing_country, billing_postal_code, total)**
  Contiene información de las compras realizadas por un cliente: el identificador de la compra, el identificador del cliente, la fecha en la que se realizó la compra, la información de facturación (dirección, ciudad, estado, país y código postal), y el monto total de la compra.
- **invoice_line (invoice_line_id, invoice_id, track_id, unit_price, quantity)**
  Almacena información detallada de los productos de una compra (invoice), incluyendo su identificador, el identificador de la compra asociada, el identificador de la canción comprada, el precio unitario y la cantidad de unidades.
- **artist (artist_id, name)**
  Contiene información sobre los artistas cuyas canciones se encuentran en el catálogo de la tienda. Incluye el identificador y nombre del artista.
- **album (album_id, title, artist_id)**
  Almacena datos de los álbumes a los que pertenecen las canciones del catálogo de la tienda. Incluye el identificador y el título del álbum, así como el identificador del artista.
- **playlist (playlist_id, name)**
  Guarda información de las listas de reproducción donde se encuentran las canciones del catálogo de la tienda. Incluye el identificador y nombre de la lista de reproducción.
- **playlist_track (playlist_id, track_id)**
  Permite tener un registro de las canciones que se encuentran en cada lista de reproducción. Incluye el identificador de la lista de reproducción y el de la canción.
- **media_type (media_type_id, name)**
  Almacena información sobre los tipos de medios de las canciones: el identificador y el nombre.
- **genre (genre_id, name)**
  Almacena información de los géneros musicales de las canciones: su identificador y nombre.
- **track (track_id, name, album_id, media_type_id, genre_id, composer, miliseconds, bytes, unit_price)**
  Guarda información sobre las canciones que se encuentran a la venta en la tienda, incluyendo su identificador, su nombre, el identificador del álbum al que pertenece, el identificador del tipo de medio, el identificador del género, el compositor, la duración en milisegundos, el tamaño en bytes y el precio unitario.

Restricciones de Inclusión

$\Pi_{artist\_id}(album) \subseteq \Pi_{artist\_id}(artist)$
$\Pi_{playlist\_id}(playlist\_track) \subseteq \Pi_{playlist\_id}(playlist)$
$\Pi_{media\_type\_id}(track) \subseteq \Pi_{media\_type\_id}(media\_type)$
$\Pi_{album\_id}(track) \subseteq \Pi_{album\_id}(album)$
$\Pi_{track\_id}(invoice\_line) \subseteq \Pi_{track\_id}(track)$
$\Pi_{customer\_id}(invoice) \subseteq \Pi_{customer\_id}(customer)$
$\Pi_{support\_rep\_id}(customer) \subseteq \Pi_{employee\_Id}(employee)$
$\Pi_{reports\_to}(employee) \subseteq \Pi_{employee\_Id}(employee)$
$\Pi_{track\_id}(playlist\_track) \subseteq \Pi_{track\_id}(track)$
$\Pi_{genre\_id}(track) \subseteq \Pi_{genre\_id}(genre)$
$\Pi_{invoice\_id}(invoice\_line) \subseteq \Pi_{invoice\_id}(invoice)$

En el repositorio se encuentran disponibles *scripts* para crear las tablas (*esquema.sql*) y las restricciones de integridad (*restricciones.sql*), así como cargar un conjunto de datos (*datos.sql*).

Para la creación de la base y la carga de los datos se recomienda seguir los siguientes pasos en la línea de comandos:

1. Crear la base de datos (*createdb -U postgres_username -E UTF8 tarea2*).
2. Ingresar a la base creada en el paso anterior (*psql -U postgres_username -d tarea2*).
3. Setear el encoding a utf8 (*set client_encoding to utf8;*).
   Para corroborar que el encoding es correcto se puede correr el comando *show client_encoding;*.
   Observación: Ambos comandos llevan el caracter ";" al final.
4. Correr los 3 scripts de carga (*\i path/to/script.sql*), en el siguiente orden:
   a) *esquema.sql*
   b) *restricciones.sql*
   c) *datos.sql*

4. Restricciones generales

1. Respetar los nombres de atributos indicados en la sección "Esquema del resultado" de cada consulta y el orden de los mismos. Las soluciones que no las respeten serán consideradas incorrectas.
2. No utilizar vistas, a menos que se indique en la letra.
3. No utilizar (SELECT ... FROM) en la lista de atributos de un SELECT.
4. No usar subconsultas en el FROM en ningún nivel de SELECT.
5. Las únicas funciones de agregación que se pueden usar son MIN, MAX, COUNT y AVG
6. Para cualquier otra cláusula que pretenda utilizar que no sea de la forma básica (SELECT ... FROM ... WHERE ... GROUP BY ... HAVING ... ORDER BY ...), consultar si está permitido su uso.

5. Consultas

A continuación se formula un conjunto de consultas. Se pide resolver las consultas 1 a 9 mediante expresiones **SQL**.

CONSULTA 1.
Devolver el conjunto de los géneros musicales de la artista "Amy Winehouse".
Esquema del resultado: name.

CONSULTA 2.
Devolver el nombre de los artistas con un único género musical.
Esquema del resultado: name.

CONSULTA 3.
Devolver el nombre, la duración (en minutos) y el precio de la canción con mayor duración comprada por clientes de Canadá.
Esquema del resultado: name, duration, unit_price.

CONSULTA 4.
Devolver el nombre y apellido de los clientes que compraron al menos 2 canciones distintas del género musical "Rock".
Esquema del resultado: first_name, last_name.

CONSULTA 5.
Devolver el nombre y apellido de los clientes que compraron todas las canciones de la artista "Amy Winehouse".
Esquema del resultado: first_name, last_name.

CONSULTA 6.
Devolver el nombre, apellido y cantidad de compras de cada cliente asesorado por "Steve Johnson".
Esquema del resultado: first_name, last_name, number_of_purchases.

CONSULTA 7.
Devolver el nombre del género musical con más cantidad de playlists.
Esquema del resultado: name.

CONSULTA 8.
Devolver el nombre y la cantidad de ventas para los álbumes con las 3 mayores cantidades de ventas. La cantidad de ventas se debe devolver en una columna denominada "number_of_purchases". Ordenar el resultado según la cantidad de ventas de forma decreciente.
Esquema del resultado: title, number_of_purchases.

CONSULTA 9.
Devolver el identificador, nombre y apellido del empleado que asesoró la mayor cantidad de ventas en el 2024
Esquema del resultado: employee_id, first_name, last_name.

CONSULTA 10.
Describa qué es lo que realiza la siguiente consulta. Justifique

```
SELECT *, COUNT(t.track_id) as count
FROM invoice_line il
    JOIN invoice i ON i.invoice_id = il.invoice_id
    JOIN track t ON t.track_id = il.track_id
WHERE i.invoice_date BETWEEN '2013-01-01' and '2013-12-31'
GROUP BY t.track_id
ORDER BY count DESC
```