

<b>EVALUACIÓN</b>	Obligatorio 2	<b>FECHA</b>	Oct. 2019
<b>MATERIA</b>	Bases de Datos 2		
<b>CARRERA</b>	Analista en Tecnologías de Información / Analista Programador		
<b>CONDICIONES</b>	<p>- <b>Entrega:</b> 14 de noviembre. <b>Hasta las 21 hs.</b></p> <p>- <b>Puntos:</b> <u>Máximo:</u> 25</p> <p>- <b>Defensa:</b> a acordar con cada docente.</p> <p>LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF.</p> <p><b>IMPORTANTE:</b></p> <ul style="list-style-type: none"> <li>- Inscribirse</li> <li>- Formar grupos de hasta dos personas.</li> <li>- Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO"</li> </ul>		

Una aplicación de streaming de música cuenta dentro de su esquema de datos con la siguiente estructura de tablas:

artista(artistaId, artistaNombre, artistaFoto, esNacional, artistaCantReproducciones)

album(albumId, albumNombre, *artistaId*, albumCantReproducciones)

cancion(cancionId, cancionNombre, *albumId*, cancionCantReproducciones)

plan(planId, planNombre, planCosto, esPlanVigente, esRecurrente)

usuario(usuariold, usuarioNombre, usuarioMail, fechaCreacion, *planId*, fechaPlanActivoDesde)

compra(comprald, *usuariold*, *planId*, fechaGenerada, fechaPaga)

playList(playListId, playListNombre, *usuariold*, playListFechaCreacion, esPlayListCurada)

playListCancion(playListId, cancionId)

historialCancion(historialCancionId, historialCancionFecha, *usuariold*, *cancionId*)

### Se Pide:

1. Crear procedimientos o funciones según corresponda para:

- Crear un procedimiento almacenado 'reproduccionesPorUsuarioPorAnio' que reciba como parámetros un año y un id de usuario, y devuelva por parámetro: la cantidad de artistas distintos escuchados por el usuario en el año, la cantidad de álbumes distintos escuchados por el usuario en el año y la cantidad de temas distintos escuchados por el usuario en el año.
- Crear un procedimiento almacenado 'albumMasEscuchadoArtista', que dado el código de un artista, devuelva por parámetro, el álbum con más reproducciones del artista, y la cantidad de reproducciones de dicho álbum.
- Implementar una función 'reproduccionesPorArtistaPorAnio', que reciba como parámetros el id de un artista y un año, devolviendo la cantidad de reproducciones del artista en el año.

- d. Implementar un procedimiento almacenado 'resumenArtistaPorAnio', que dados el código de un artista y un año, devuelva por parámetro: la cantidad de reproducciones totales que tuvo el artista en el año, la cantidad de playLists distintas creadas en el año proporcionado en que aparece alguno de sus temas, la cantidad de usuarios distintos que escucharon alguno de sus temas en el año.
- e. Crear una función, 'artistaNacionalMasEscuchadoRangoFechas' que reciba por parámetros un rango de fechas y devuelva el artista nacional más escuchado en dicho período.
- f. Implementar un procedimiento almacenado, 'infoAlbum', que reciba por parámetro el id de un álbum y devuelva, también por parámetros: la cantidad de temas del álbum, la fecha en que se reprodujo por primera vez alguno de los temas del álbum y la cantidad de reproducciones del álbum.
- g. Crear una función 'temasPorArtista' que reciba como parámetro un id de artista y devuelva la cantidad de temas de dicho artista.

**(7 puntos)**

- 2. Crear disparadores necesarios para realizar las siguientes actividades (considerar acciones múltiples para la implementación de los triggers):
  - a. Cada vez que se ingrese un registro en historialCanción, se actualicen la cantidad de reproducciones de la canción, el álbum y el artista.
  - b. Cuando se ingrese una playList, validar que no tenga usuario en caso de que sea curada y que lo tenga en caso contrario.
  - c. Controlar que las playLists no tengan más de 100 canciones.
  - d. No permitir asignar planes no vigentes a los usuarios.
  - e. Implementar un disparador que al eliminar un artista, elimine todo lo relacionado al mismo (álbum, canción, sacarlo de playlist y de historial).

**(10 puntos)**

- 3. Resolver mediante consultas SQL:
  - a. Mostrar los usuarios que han escuchado todas las playlists curadas (basta con haber escuchado con un tema de la playlist para considerar escuchó la playlist).
  - b. Mostrar para cada artista, la cantidad de álbumes que tiene con más de 10 temas. En caso de que haya artistas sin álbumes, también deben mostrarse.
  - c. Devolver id y nombre del/los usuario/s que escuchó/escucharon más temas nacionales en lo que va del año.
  - d. Devolver el id y nombre de los usuarios, que tengan más de 3 playlists, que hayan escuchado más de 10 temas por mes en los últimos 3 meses y que no hayan escuchado más de 20 artistas distintos en el mes actual.
  - e. Devolver el nombre de los usuarios que hayan escuchado temas de todos los álbumes de 'Jaime Roos'.

- 
- f. Eliminar los artistas que no tengan temas en playlists y que no tengan reproducciones en lo que va del año.

**(6 puntos)**

**4. Vistas:**

- a. Crear una vista 'usuariosPorPlan' que muestre por plan, la cantidad de usuarios que tiene, considerando solamente los planes con más de 10 usuarios.
- b. Crear una vista, 'promedioReproduccionesUsuariosMesAnio' que muestra para cada mes, año el promedio de reproducciones (reproducciones totales / usuarios distintos). Mostrar en el resultado solo los meses/años con más de 100 reproducciones.

**(2 puntos)**

**Se debe entregar:**

- a. Script con los procedimientos y funciones.
- b. Script con los disparadores implementados.
- c. Script con consultas y vistas.
- d. Informe de consideraciones de la solución.

**Consideraciones generales:**

1. Los docentes de la materia cumplirán el rol de usuario final del producto a los efectos de evacuar las dudas que puedan surgir a los estudiantes en detalles que no estén incluidos explícitamente en la letra. Independientemente de esto, los alumnos podrán investigar sobre sistemas existentes, así como aportes basados en su propia experiencia o relevamiento con terceros para enriquecer la solución a los problemas planteados siempre que no contradiga lo explicitado en la letra. Cualquier agregado deberá documentarse claramente en la solución y será considerado positivamente en la evaluación. Modificaciones de la letra que puedan surgir durante el curso, serán publicadas en aulas y deberán considerarse en la entrega final.
2. La corrección del obligatorio se hará en base a la estructura entregada junto con la letra del mismo, por lo que los puntos desarrollados deben ser testeados sobre esta estructura. Soluciones a los puntos del obligatorio que no ejecuten correctamente sobre la estructura proporcionada serán evaluados como incorrectos.
3. Durante la última semana los docentes no contestarán dudas del Obligatorio por ningún medio. Esta consideración intenta evitar que los alumnos dejen la implementación del obligatorio para último momento. Se insta a los estudiantes a desarrollar el obligatorio durante el transcurso del semestre para entregar un trabajo de calidad.