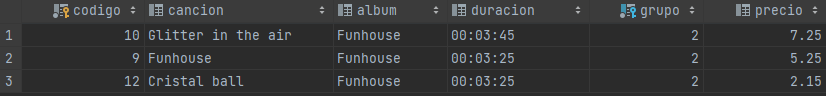
**Simulacro Examen**

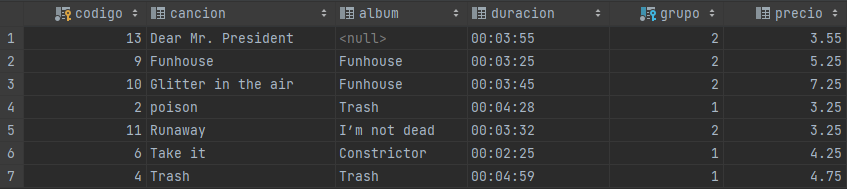
1. Muestra las canciones del grupo 2 cuyo álbum empiece por F, mostrando los resultados por canción de forma descendente:

SELECT c.\*  
FROM canciones c  
WHERE c.grupo = 2  
 AND c.album LIKE 'F%'  
ORDER BY c.cancion DESC



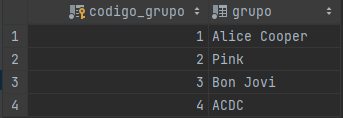
1. Muestra las canciones que duren entre 2 y 3 minutos y/o que cuesten más de 3 euros, ordenando los resultados por canción ascendente y por álbum descendente:

SELECT c.\*  
FROM canciones c  
WHERE c.duracion BETWEEN '00:02:00' AND '00:03:00'  
 OR c.precio > 3  
ORDER BY c.cancion, c.album DESC



1. Inserta estos dos grupos: codigo: 3, grupo: Bon Jovi; código:4, grupo: ACDC

INSERT INTO grupo  
VALUES (3, 'Bon Jovi'),  
 (4, 'ACDC')



1. Inserta estas canciones de Bon Jovi:
   1. codigo: 14, canción: Livin on a player, álbum: Sleepery when wet, duración: 0:4:11, precio: 3,70
   2. codigo: 15, canción: Never say goodbye, álbum: Sleepery when wet, duración: 0:4:50, precio: 4,70

INSERT INTO canciones  
VALUES (14, 'Livin on a player', 'Sleepery when wet', '00:04:11', 3, 3.7),  
 (15, 'Never say goodbye', 'Sleepery when wet', '00:04:50', 3, 4.7)



1. Inserta estas canciones de ACDC:
   1. código: 16, canción: Highway to Hell, álbum: ACDC Live, duración: 0:5:45, precio: 4,50
   2. código: 17, canción: Thunderstruck, álbum: The Razors Edge, duración: 0:5:50, precio: 7,25

INSERT INTO canciones  
VALUES (16, 'Highway to Hell', 'ACDC Live', '00:05:45', 4, 4.5),  
 (17, 'Thunderstruck', 'The Razors Edge', '00:05:50', 4, 7.25)



1. Ponle a la canción 7 una duración de 0:3:25

UPDATE canciones c  
SET c.duracion = '00:03:25'  
WHERE c.codigo = 7



1. Cambia el nombre del álbum” Greatest Hits” por “Greatest Hits… so far!!!”:

ALTER TABLE canciones  
 MODIFY album VARCHAR(50)

UPDATE canciones  
SET album = 'Greatest Hits… so far!!!'  
WHERE album = 'Greatest Hits'



1. Elimina la canción Cristal Ball:

DELETE  
FROM canciones  
WHERE cancion = 'Cristal ball'



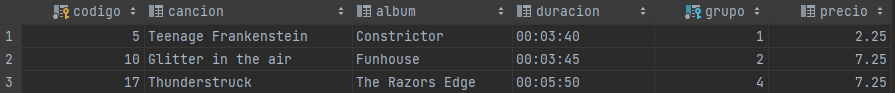
1. Muestra la cantidad de canciones que hay de cada grupo:

SELECT c.grupo AS 'Codigo\_Grupo', g.grupo AS 'Nombre\_Grupo', COUNT(c.codigo) AS 'Cantidad\_canciones'  
FROM canciones c  
 JOIN grupo g on g.codigo\_grupo = c.grupo  
GROUP BY c.grupo



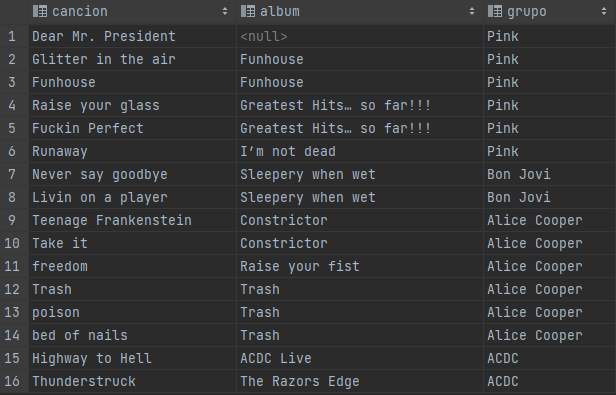
1. Muestra toda la información de las canciones que tengan el precio máximo y el precio mínimo:

SELECT c.\*  
FROM canciones c  
WHERE c.precio = (SELECT MAX(c.precio)  
 FROM canciones c)  
 OR c.precio = (SELECT MIN(c.precio)  
 FROM canciones c)



1. Muestra la canción, el álbum y el nombre del grupo, ordenando los resultados por grupo descendente, álbum ascendente y canción descendente:

SELECT c.cancion, c.album, g.grupo  
FROM canciones c  
 JOIN grupo g on g.codigo\_grupo = c.grupo  
ORDER BY g.grupo DESC, c.album, c.cancion DESC



1. Muestra el total del precio de cada álbum que no sea nulo, considerando solo los totales que sean mayores de 5 euros:

SELECT c.album, SUM(c.precio) AS 'precio\_total'  
FROM canciones c  
GROUP BY c.album  
HAVING c.album IS NOT NULL AND precio\_total > 5

