

Base de datos Mellon Music

```
1 CREATE DATABASE IF NOT EXISTS mellon;
2 USE mellon;
3
4 CREATE TABLE IF NOT EXISTS mellon.artist (
5     ArtistID INT NOT NULL AUTO_INCREMENT,
6     Name VARCHAR(255) NOT NULL,
7     Genre VARCHAR(100) NULL DEFAULT NULL,
8     ArtistID INT NULL DEFAULT NULL,
9     PRIMARY KEY (ArtistID),
10    INDEX (ArtistID ASC) VISIBLE,
11    FOREIGN KEY (FK_songID)
12        REFERENCES mellon.album (AlbumID)
13        ON DELETE CASCADE
14        ON UPDATE CASCADE
15 );
16
17 SELECT * FROM mellon.artist;
18     INSERT INTO artist VALUES (null, 'Intoxicados','punk');
19     INSERT INTO artist VALUES (null, 'Patricio Rey y los redondos','rock');
20     INSERT INTO artist VALUES (null, 'Los Tipitos','rock');
21     INSERT INTO artist VALUES (null, 'Bob Marley','reagge');
22     INSERT INTO artist VALUES (null, 'Led Zeppelin','rock');
23
24
25 CREATE TABLE IF NOT EXISTS mellon.album (
26     AlbumID INT NOT NULL AUTO_INCREMENT,
27     Title VARCHAR(255) NOT NULL,
28     ReleaseYear VARCHAR(100) NULL DEFAULT NULL,
29     ArtistID INT NULL DEFAULT NULL,
30     PRIMARY KEY (AlbumID),
31     INDEX (ArtistID ASC) VISIBLE,
32     FOREIGN KEY (ArtistID)
33         REFERENCES mellon.artist (ArtistID)
34         ON DELETE CASCADE
35         ON UPDATE CASCADE
36 );
37
```

```
38  SELECT * FROM mellon.album;
39
40  INSERT INTO Album VALUES (null,'Otro día en el planeta Tierra', '2005', null);
41  INSERT INTO album VALUES (null, 'Oktubre', '2008', null);
42  INSERT INTO Album VALUES (null,'Armando Camaleon', '2004', null);
43  INSERT INTO Album VALUES (null,'Exodus', '1977', null);
44  INSERT INTO Album VALUES (null,'Led Zeppelin', '1969', null);
45
46
47  CREATE TABLE IF NOT EXISTS mellon.user (
48      UserID INT NOT NULL AUTO_INCREMENT,
49      Username VARCHAR(50) NOT NULL,
50      Email VARCHAR(255) NULL NULL,
51      Password VARCHAR(255) NOT NULL,
52      RegistrationDate DATE NULL DEFAULT NULL,
53      PRIMARY KEY (UserID),
54      UNIQUE INDEX Email (Email ASC) VISIBLE,
55      FOREIGN KEY (ArtistID)
56          REFERENCES mellon.artist (ArtistID)
57          ON DELETE CASCADE
58          ON UPDATE CASCADE
59  );
60  SELECT * FROM mellon.user;
61
62  INSERT INTO Album VALUES (null,'Val', 'flash@a.com', '123', null);
63  INSERT INTO Album VALUES (null,'Luk', 'mood@a.com', '123', null);
64  INSERT INTO Album VALUES (null,'Fran', 'rain@a.com', '123', null);
65  INSERT INTO Album VALUES (null,'Num', 'cloud@a.com', '123', null);
```


```
66
67
68 CREATE TABLE IF NOT EXISTS mellon.song (
69     SongID INT NOT NULL AUTO_INCREMENT,
70     Title VARCHAR(255) NOT NULL,
71     Length TIME NULL DEFAULT NULL,
72     AlbumID INT NULL DEFAULT NULL,
73     PRIMARY KEY (SongID),
74     INDEX (AlbumID ASC) VISIBLE,
75     CONSTRAINT song_ibfk_1
76     FOREIGN KEY (AlbumID)
77     REFERENCES mellon.album (AlbumID)
78     ON DELETE CASCADE
79     ON UPDATE CASCADE
80 );
81
82 SELECT * FROM mellon.song;
83
84 INSERT INTO song VALUES (null, 'Fuego', '00:04:30', null);
85 INSERT INTO song VALUES (null, 'Jijiji', '00:06:03', null);
86 INSERT INTO song VALUES (null, 'Brujeria', '00:03:10', null);
87 INSERT INTO song VALUES (null, 'Natural Mystic', '00:03:27', null);
88 INSERT INTO song VALUES (null, 'Good Times Bad Times', '00:02:46', null);
89
90
91 CREATE TABLE IF NOT EXISTS mellon.playlist (
92     PlaylistID INT NOT NULL AUTO_INCREMENT,
93     Name VARCHAR(50) NOT NULL,
94     UserID INT NULL DEFAULT NULL,
95     PRIMARY KEY (PlaylistID),
96     INDEX UserID (UserID ASC) VISIBLE,
97     CONSTRAINT playlist_ibfk_1
98     FOREIGN KEY (UserID)
99     REFERENCES mellon.user (UserID),
100     CONSTRAINT artist
```

```

100     CONSTRAINT artist
101         FOREIGN KEY (PlaylistID)
102         REFERENCES mellon.artist (Artist)
103         ON DELETE NO ACTION
104         ON UPDATE NO ACTION,
105     CONSTRAINT song
106         FOREIGN KEY (PlaylistID)
107         REFERENCES mellon.song (SongID)
108         ON DELETE NO ACTION
109         ON UPDATE NO ACTION
110 );
111
112
113 SELECT * FROM mellon.playlist;
114
115 INSERT INTO playlist VALUES (null,'Lokura', null);
116 insert into playlist values (null, 'Fiesta', null);
117 insert into playlist values (null, 'Dormir', null);
118 INSERT INTO playlist VALUES (null,'Tranqui', null);

```

Agrupar mediante GROUP BY con una funcion agregada

Code Blame 3 lines (3 loc) · 83 Bytes  Code 55% faster v

```

1     select Genre AS "Genero", count(*) AS "Total Bandas"
2     from artist
3     group by Genre

```

Agrupar mediante GROUP BY

Code Blame 2 lines (2 loc) · 42 Bytes

```

1     select Genre from artist
2     group by Genre

```

Eliminar tablas

```
Code Blame 3 lines (3 loc) · 59 Bytes

1 DROP TABLE genre;
2 DROP TABLE account;
3 DROP TABLE megusta ;
```

FK_PLAYLIST

```
Code Blame 15 lines (15 loc) · 394 Bytes

1 ALTER TABLE mellon.playlist
2 ADD CONSTRAINT playlist_ibfk_1
3 FOREIGN KEY (UserID)
4 REFERENCES mellon.user (UserID),
5 ADD CONSTRAINT artist
6 FOREIGN KEY (PlaylistID)
7 REFERENCES mellon.artist (ArtistID)
8 ON DELETE CASCADE
9 ON UPDATE CASCADE,
10 ADD CONSTRAINT song
11 FOREIGN KEY (PlaylistID)
12 REFERENCES mellon.song (SongID)
13 ON DELETE CASCADE
14 ON UPDATE CASCADE
15 ;
```

FK_ALBUM

```
Code Blame 6 lines (6 loc) · 147 Bytes

1 ALTER TABLE mellon.album
2 ADD CONSTRAINT FK_ArtistID
3 FOREIGN KEY (ArtistID)
4 REFERENCES mellon.artist (ArtistID)
5 ON DELETE CASCADE
6 ON UPDATE CASCADE;
```

FK_SONG

Code Blame 6 lines (6 loc) · 141 Bytes

```
1 ALTER TABLE mellon.song
2 ADD CONSTRAINT FK_songID
3 FOREIGN KEY (AlbumID)
4 REFERENCES mellon.album (AlbumID)
5 ON DELETE CASCADE
6 ON UPDATE CASCADE;
```

Base de datos 1

```
1 -- Crear la base de datos si no existe
2 CREATE DATABASE IF NOT EXISTS mellon;
3 USE mellon;
4
5 -- Crear la tabla artist
6 CREATE TABLE IF NOT EXISTS artist (
7     ArtistID INT NOT NULL AUTO_INCREMENT,
8     Name VARCHAR(255) NOT NULL,
9     Genre VARCHAR(100) NULL DEFAULT NULL,
10    PRIMARY KEY (ArtistID)
11 );
12
13 -- Insertar datos en la tabla artist
14 INSERT INTO artist (Name, Genre) VALUES ('Intoxicados', 'punk');
15 INSERT INTO artist (Name, Genre) VALUES ('Patricio Rey y los redondos', 'rock');
16 INSERT INTO artist (Name, Genre) VALUES ('Los Tipitos', 'rock');
17 INSERT INTO artist (Name, Genre) VALUES ('Bob Marley', 'reggae');
18 INSERT INTO artist (Name, Genre) VALUES ('Led Zeppelin', 'rock');
19
20 -- Crear la tabla album
21 CREATE TABLE IF NOT EXISTS album (
22     AlbumID INT NOT NULL AUTO_INCREMENT,
23     Title VARCHAR(255) NOT NULL,
24     ReleaseYear VARCHAR(100) NULL DEFAULT NULL,
25     ArtistID INT NULL DEFAULT NULL,
26     PRIMARY KEY (AlbumID),
27     INDEX (ArtistID ASC) VISIBLE,
28     FOREIGN KEY (ArtistID)
29     REFERENCES artist (ArtistID)
30     ON DELETE CASCADE
31     ON UPDATE CASCADE
32 );
33
34 -- Insertar datos en la tabla album
35 INSERT INTO album (Title, ReleaseYear, ArtistID) VALUES ('Otro dia en el planeta Tierra', '2005', 1);
36 INSERT INTO album (Title, ReleaseYear, ArtistID) VALUES ('Oktubre', '2008', 2);
37 INSERT INTO album (Title, ReleaseYear, ArtistID) VALUES ('Armando Camaleon', '2004', 3);
38 INSERT INTO album (Title, ReleaseYear, ArtistID) VALUES ('Exodus', '1977', 4);
39 INSERT INTO album (Title, ReleaseYear, ArtistID) VALUES ('Led Zeppelin', '1969', 5);
40
```

```

41  -- Crear la tabla user
42  CREATE TABLE IF NOT EXISTS user (
43      UserID INT NOT NULL AUTO_INCREMENT,
44      Username VARCHAR(50) NOT NULL,
45      Email VARCHAR(255) NULL,
46      Password VARCHAR(255) NOT NULL,
47      RegistrationDate DATE NULL DEFAULT NULL,
48      PRIMARY KEY (UserID),
49      UNIQUE INDEX Email (Email ASC) VISIBLE
50  );
51
52  -- Insertar datos en la tabla user
53  INSERT INTO user (Username, Email, Password, RegistrationDate) VALUES ('Val', 'flash@a.com', '123', CURDATE());
54  INSERT INTO user (Username, Email, Password, RegistrationDate) VALUES ('Luk', 'mood@a.com', '123', CURDATE());
55  INSERT INTO user (Username, Email, Password, RegistrationDate) VALUES ('Fran', 'rain@a.com', '123', CURDATE());
56  INSERT INTO user (Username, Email, Password, RegistrationDate) VALUES ('Num', 'cloud@a.com', '123', CURDATE());
57
58  -- Crear la tabla song
59  CREATE TABLE IF NOT EXISTS song (
60      SongID INT NOT NULL AUTO_INCREMENT,
61      Title VARCHAR(255) NOT NULL,
62      Length TIME NULL DEFAULT NULL,
63      AlbumID INT NULL DEFAULT NULL,
64      PRIMARY KEY (SongID),
65      INDEX (AlbumID ASC) VISIBLE,
66      FOREIGN KEY (AlbumID)
67          REFERENCES album (AlbumID)
68          ON DELETE CASCADE
69          ON UPDATE CASCADE
70  );
71
72  -- Insertar datos en la tabla song
73  INSERT INTO song (Title, Length, AlbumID) VALUES ('Fuego', '00:04:30', 1);
74  INSERT INTO song (Title, Length, AlbumID) VALUES ('Jijiji', '00:06:03', 2);
75  INSERT INTO song (Title, Length, AlbumID) VALUES ('Brujeria', '00:03:10', 3);
76  INSERT INTO song (Title, Length, AlbumID) VALUES ('Natural Mystic', '00:03:27', 4);
77  INSERT INTO song (Title, Length, AlbumID) VALUES ('Good Times Bad Times', '00:02:46', 5);

```

```

78
79  -- Crear la tabla playlist
80  CREATE TABLE IF NOT EXISTS playlist (
81      PlaylistID INT NOT NULL AUTO_INCREMENT,
82      Name VARCHAR(50) NOT NULL,
83      UserID INT NULL DEFAULT NULL,
84      PRIMARY KEY (PlaylistID),
85      INDEX (UserID ASC) VISIBLE,
86      FOREIGN KEY (UserID)
87          REFERENCES user (UserID)
88  );
89
90  -- Insertar datos en la tabla playlist
91  INSERT INTO playlist (Name, UserID) VALUES ('Lokura', 1);
92  INSERT INTO playlist (Name, UserID) VALUES ('Fiesta', 2);
93  INSERT INTO playlist (Name, UserID) VALUES ('Dormir', 3);
94  INSERT INTO playlist (Name, UserID) VALUES ('Tranqui', 4);

```

Operador logico AND

```
Code Blame 7 lines (3 loc) · 105 Bytes Code 55% f

1  SELECT * FROM mellon.artist;
2
3  select * from artist
4      where ArtistID >= 0 and Genre = "Rock"
5
6
7
```

Operador logico NOT

```
Code Blame 4 lines (3 loc) · 76 Bytes

1  SELECT * FROM mellon.artist;
2
3  select * from artist
4      where not Genre = "Rock"
```

Operador logico OR

```
Code Blame 4 lines (3 loc) · 92 Bytes Code 55% f

1  SELECT * FROM mellon.artist;
2
3  select * from artist
4      where Genre ="Reggae" or Genre = "Rock"
```

UPDATE ALBUM

```
Code Blame 11 lines (10 loc) · 221 Bytes Code 55% f

1  SELECT * FROM mellon.album;
2
3  UPDATE album
4  SET title = 'Otro dia en el planeta Tierra'
5  WHERE albumid = 1;
6  UPDATE album
7  SET title = 'Oktubre'
8  WHERE albumid = 2;
9  UPDATE album
10 SET title = 'Armando Camaleon'
11 WHERE albumid = 3;
```


INNER JOIN

```
Code Blame 4 lines (4 loc) · 157 Bytes Code 55% faster wi

1  SELECT * FROM song
2  INNER JOIN album ON song.albumID = album.albumID
3  INNER JOIN artist ON album.artistID = artist.artistID
4  WHERE artist.name = 'Led Zeppelin';
```

Consulta que especifica sobre las canciones que pertenecen a un álbum particular ("Led Zeppelin"), junto con los detalles del álbum y el artista asociados.

INNER JOIN CON FILTROS

1

```
1  SELECT album.*, artist.name AS artist_name
2
3  FROM album
4
5  INNER JOIN artist ON album.artistID = artist.artistID
6
7  WHERE artist.name = 'Led Zeppelin';
```

Esta consulta permite obtener los detalles de los álbumes del artista "Led Zeppelin" utilizando un INNER JOIN para combinar la información de las tablas album y artist

2

```
1  SELECT song.Title AS SongTitle, album.Title AS AlbumTitle, artist.Name AS ArtistName, album.ReleaseYear
2
3  FROM song
4
5  INNER JOIN album ON song.AlbumID = album.AlbumID
6
7  INNER JOIN artist ON album.ArtistID = artist.ArtistID
8
9  WHERE album.ReleaseYear > 2000;
```

Esta consulta obtiene detalles de las canciones de álbumes de un artista específico cuyo nombre sea "Led Zeppelin"

Insert artist

```
1  SELECT * FROM mellon.artist;
2
3  INSERT INTO artist VALUES (null, 'Intoxicados','punk');
4  INSERT INTO artist VALUES (null, 'Patricio Rey y los redondos','rock');
5  INSERT INTO artist VALUES (null, 'Los Tipitos','rock');
6  INSERT INTO artist VALUES (null, 'Bob Marley','reagge');
7  INSERT INTO artist VALUES (null, 'Led Zeppelin','rock')
```

Insert album

```
1  SELECT * FROM mellon.album;
2
3  INSERT INTO Album VALUES (null,'Exodus', '1977', null);
4  insert into album values (null, 'Led Zeppelin', '1969', null);
```

Insert playlist

```
1  SELECT * FROM mellon.playlist;
2
3  INSERT INTO playlist VALUES (null,'Lokura', null);
4  insert into playlist values (null, 'Fiesta', null);
5  insert into playlist values (null, 'Dormir', null);
6  INSERT INTO playlist VALUES (null,'Tranqui', null);
```

Insert song

```
1  SELECT * FROM mellon.song;
2
3  INSERT INTO song VALUES (null, 'Fuego', '00:04:30', null);
4  INSERT INTO song VALUES (null, 'Jijiji', '00:06:03', null);
5  INSERT INTO song VALUES (null, 'Brujeria', '00:03:10', null);
6  INSERT INTO song VALUES (null, 'Natural Mystic', '00:03:27', null);
7  INSERT INTO song VALUES (null, 'Good Times Bad Times', '00:02:46', null);
```

Insert user

```
1  SELECT * FROM mellon.user;
2
3  INSERT INTO user VALUES (null, 'Val', 'flash@a.com',123,null);
4  INSERT INTO user VALUES (null, 'Luk', 'mood@a.com',123,null);
5  INSERT INTO user VALUES (null, 'Fran', 'rain@a.com',123,null);
6  INSERT INTO user VALUES (null, 'Num', 'cloud@a.com',123,null);
```

WHERE

1

```
1  SELECT * FROM Album WHERE ArtistID = ( SELECT ArtistID FROM Artist WHERE Name = 'Bob Marley' );
```

Esta consulta logra obtener todos los registros de la tabla Album donde el ArtistID corresponde al artista llamado "Bob Marley". La subconsulta selecciona el ArtistID de la tabla Artist para el artista con el nombre 'Bob Marley', y luego la consulta principal utiliza este ArtistID para filtrar los álbumes correspondientes.

2

```
1  SELECT Song.* FROM Song JOIN Album ON Song.AlbumID = Album.AlbumID JOIN Artist ON Album.ArtistID = Artist.ArtistID WHERE Artist.Name = 'Bob Marley';
```

Esta consulta funcionará para obtener todas las canciones de la tabla Song que pertenecen a álbumes del artista llamado 'Bob Marley'. La consulta une las tablas Song, Album y Artist para filtrar las canciones basándose en el nombre del artista.

WHERE LIMIT

```
1  SELECT *
2  FROM Album
3  WHERE ArtistID = 1
4  LIMIT 5;
```

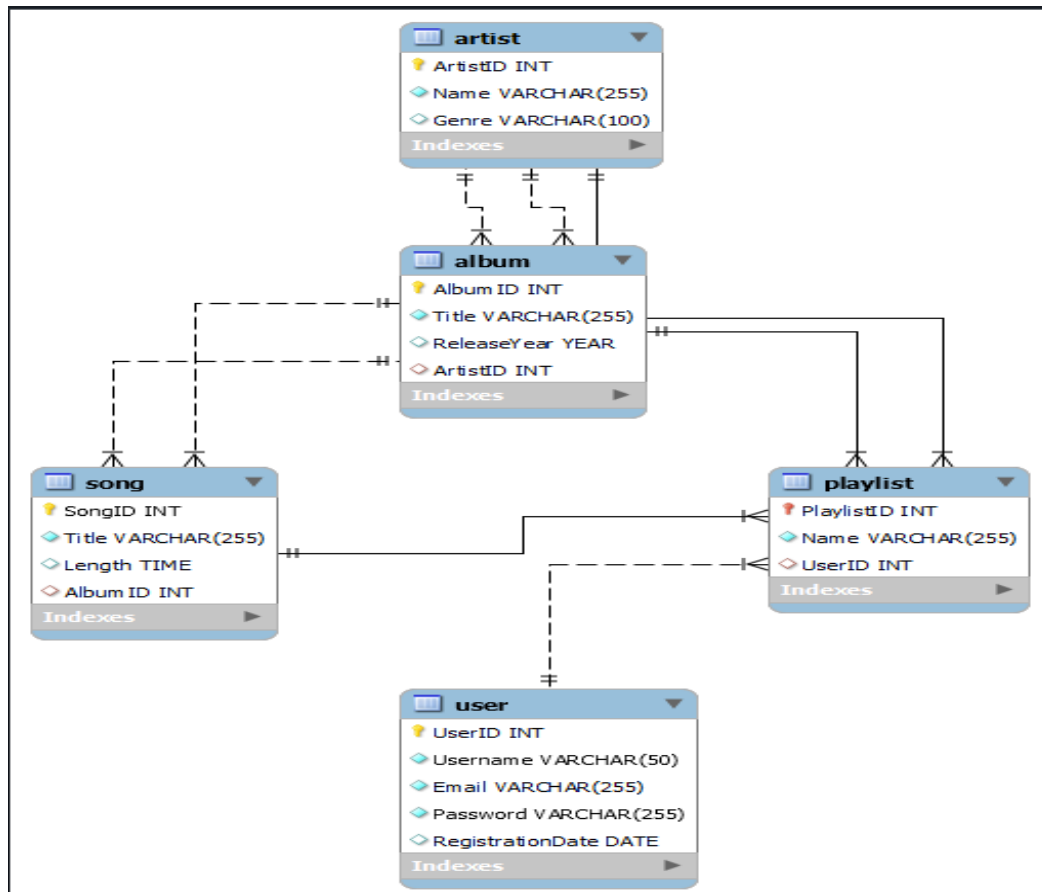
El objetivo de esta consulta es obtener los primeros 5 álbumes que pertenecen al artista con ArtistID igual a 1.

WHERE BETWEEN

```
1  SELECT *
2  FROM Playlist
3  WHERE CreationDate BETWEEN '2023-01-01' AND '2023-12-31';
```

Esta consulta funciona para seleccionar playlist creadas en unas fechas específicas. La columna CreationDate se usa para almacenar la fecha de creación de cada playlist.

CROW FOOT



DER

