

Mellon Music



Integrantes:

Numa De Agostini

Valentina Pich

Ledesma Lucas

Francisco Oviedo

Fecha: 14/06/2024

Diagrama Entidad Relación

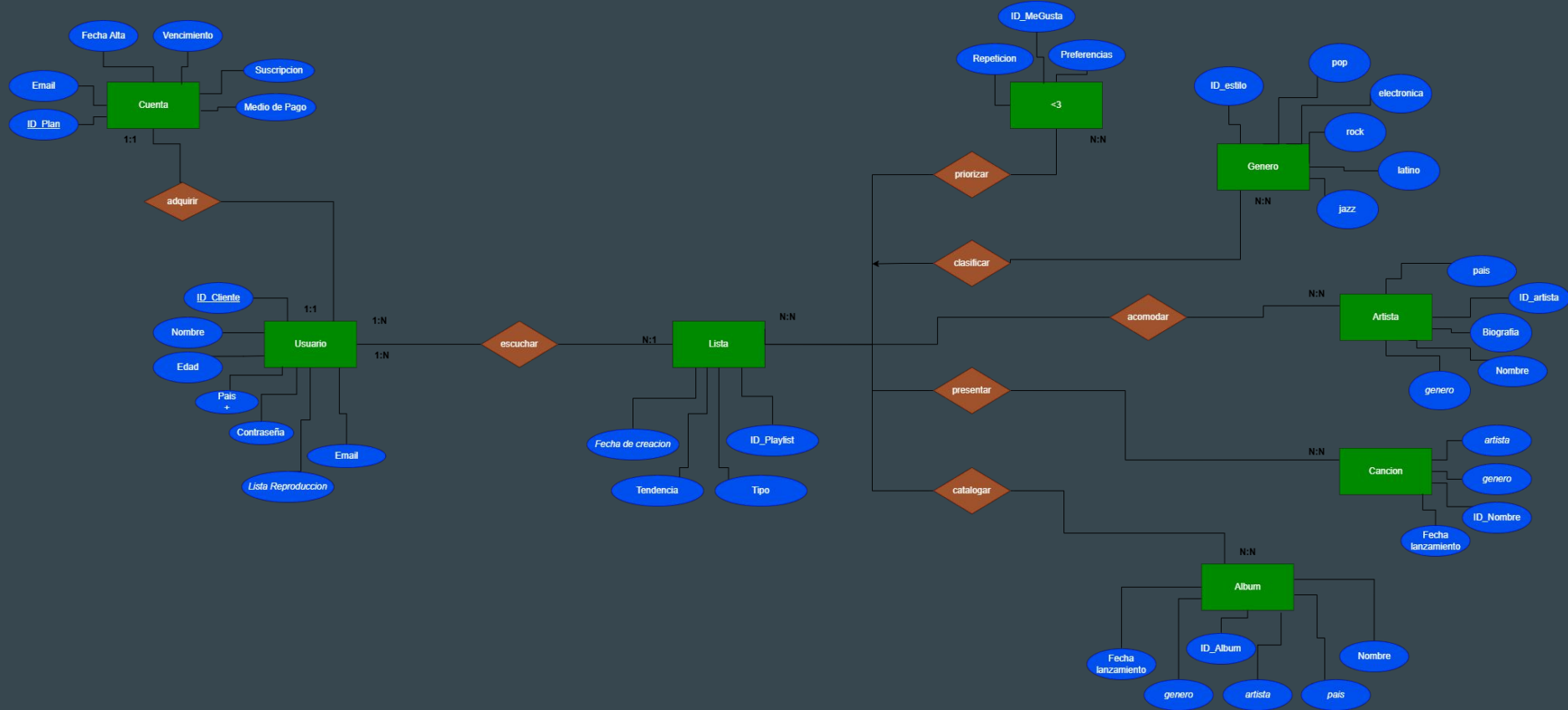
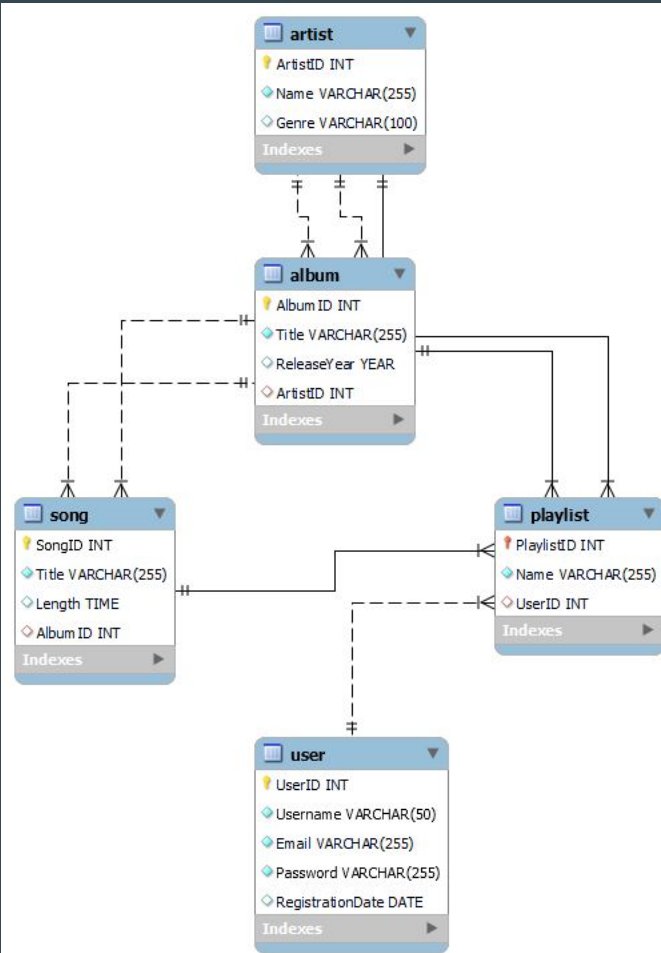


Diagrama Crow Foot



Tablas

```
CREATE DATABASE IF NOT EXISTS mellon;
USE mellon;

CREATE TABLE IF NOT EXISTS mellon.artist (
  ArtistID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(255) NOT NULL,
  Genre VARCHAR(100) NULL DEFAULT NULL,
  ArtistID INT NULL DEFAULT NULL,
  PRIMARY KEY (ArtistID),
  INDEX (ArtistID ASC) VISIBLE,
  FOREIGN KEY (FK_songID)
    REFERENCES mellon.album (AlbumID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

SELECT * FROM mellon.artist;

INSERT INTO artist VALUES (null, 'Intoxicados','punk');
INSERT INTO artist VALUES (null, 'Patricio Rey y los redondos','rock');
INSERT INTO artist VALUES (null, 'Los Tipitos','rock');
INSERT INTO artist VALUES (null, 'Bob Marley','reagge');
INSERT INTO artist VALUES (null, 'Led Zeppelin','rock');
```

```
CREATE TABLE IF NOT EXISTS mellon.song (
  SongID INT NOT NULL AUTO_INCREMENT,
  Title VARCHAR(255) NOT NULL,
  Length TIME NULL DEFAULT NULL,
  AlbumID INT NULL DEFAULT NULL,
  PRIMARY KEY (SongID),
  INDEX (AlbumID ASC) VISIBLE,
  CONSTRAINT song_ibfk_1
    FOREIGN KEY (AlbumID)
      REFERENCES mellon.album (AlbumID)
      ON DELETE CASCADE
      ON UPDATE CASCADE
);

SELECT * FROM mellon.song;

INSERT INTO song VALUES (null, 'Fuego', '00:04:30', null);
INSERT INTO song VALUES (null, 'Dijiji', '00:06:03', null);
INSERT INTO song VALUES (null, 'Brujeria', '00:03:10', null);
INSERT INTO song VALUES (null, 'Natural Mystic', '00:03:27', null);
INSERT INTO song VALUES (null, 'Good Times Bad Times', '00:02:46', null);
```

```
CREATE TABLE IF NOT EXISTS mellon.playlist (
  PlaylistID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(50) NOT NULL,
  UserID INT NULL DEFAULT NULL,
  PRIMARY KEY (PlaylistID),
  INDEX UserID (UserID ASC) VISIBLE,
  CONSTRAINT playlist_ibfk_1
    FOREIGN KEY (UserID)
      REFERENCES mellon.user (UserID),
  CONSTRAINT artist
    FOREIGN KEY (PlaylistID)
      REFERENCES mellon.artist (Artist)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT song
    FOREIGN KEY (PlaylistID)
      REFERENCES mellon.song (SongID)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION
);

SELECT * FROM mellon.playlist;

INSERT INTO playlist VALUES (null, 'Lokura', null);
insert into playlist values (null, 'Fiesta', null);
insert into playlist values (null, 'Dormir', null);
INSERT INTO playlist VALUES (null, 'Tranqui', null);
```

```
CREATE TABLE IF NOT EXISTS mellon.user (
  UserID INT NOT NULL AUTO_INCREMENT,
  Username VARCHAR(50) NOT NULL,
  Email VARCHAR(255) NULL NULL,
  Password VARCHAR(255) NOT NULL,
  RegistrationDate DATE NULL DEFAULT NULL,
  PRIMARY KEY (UserID),
  UNIQUE INDEX Email (Email ASC) VISIBLE,
  FOREIGN KEY (ArtistID)
    REFERENCES mellon.artist (ArtistID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

SELECT * FROM mellon.user;

INSERT INTO Album VALUES (null, 'Val', 'flash@a.com', '123', null);
INSERT INTO Album VALUES (null, 'Luk', 'mood@a.com', '123', null);
INSERT INTO Album VALUES (null, 'Fran', 'rain@a.com', '123', null);
INSERT INTO Album VALUES (null, 'Num', 'cloud@a.com', '123', null);
```

```
CREATE TABLE IF NOT EXISTS mellon.album (
  AlbumID INT NOT NULL AUTO_INCREMENT,
  Title VARCHAR(255) NOT NULL,
  ReleaseYear VARCHAR(100) NULL DEFAULT NULL,
  ArtistID INT NULL DEFAULT NULL,
  PRIMARY KEY (AlbumID),
  INDEX (ArtistID ASC) VISIBLE,
  FOREIGN KEY (ArtistID)
    REFERENCES mellon.artist (ArtistID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

SELECT * FROM mellon.album;

INSERT INTO Album VALUES (null, 'Otro dia en el planeta Tierra', '2005', null);
INSERT INTO album VALUES (null, 'Oktubre', '2008', null);
INSERT INTO Album VALUES (null, 'Armando Camaleon', '2004', null);
INSERT INTO Album VALUES (null, 'Exodus', '1977', null);
INSERT INTO Album VALUES (null, 'Led Zeppelin', '1969', null);
```

Tabla Artista

```
CREATE DATABASE IF NOT EXISTS mellon;
USE mellon;

CREATE TABLE IF NOT EXISTS mellon.artist (
  ArtistID INT NOT NULL AUTO_INCREMENT,
  Name VARCHAR(255) NOT NULL,
  Genre VARCHAR(100) NULL DEFAULT NULL,
  ArtistID INT NULL DEFAULT NULL,
  PRIMARY KEY (ArtistID),
  INDEX (ArtistID ASC) VISIBLE,
  FOREIGN KEY (FK_songID)
    REFERENCES mellon.album (AlbumID)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

SELECT * FROM mellon.artist;

INSERT INTO artist VALUES (null, 'Intoxicados','punk');
INSERT INTO artist VALUES (null, 'Patricio Rey y los redondos','rock');
INSERT INTO artist VALUES (null, 'Los Tipitos','rock');
INSERT INTO artist VALUES (null, 'Bob Marley','reggae');
INSERT INTO artist VALUES (null, 'Led Zeppelin','rock');
```

	ArtistID	Name	Genre
▶	1	Intoxicados	punk
	2	Patricio Rey y los redondos	rock
	3	Los Tipitos	rock
	4	Bob Marley	reggae
	5	Led Zeppelin	rock
*	NULL	NULL	NULL

Tabla Cancion

```
CREATE TABLE IF NOT EXISTS mellon.song (  
  SongID INT NOT NULL AUTO_INCREMENT,  
  Title VARCHAR(255) NOT NULL,  
  Length TIME NULL DEFAULT NULL,  
  AlbumID INT NULL DEFAULT NULL,  
  PRIMARY KEY (SongID),  
  INDEX (AlbumID ASC) VISIBLE,  
  CONSTRAINT song_ibfk_1  
  FOREIGN KEY (AlbumID)  
  REFERENCES mellon.album (AlbumID)  
  ON DELETE CASCADE  
  ON UPDATE CASCADE  
);
```

```
SELECT * FROM mellon.song;
```

```
INSERT INTO song VALUES (null, 'Fuego', '00:04:30', null);  
INSERT INTO song VALUES (null, 'Jijiji', '00:06:03', null);  
INSERT INTO song VALUES (null, 'Brujeria', '00:03:10', null);
```

	SongID	Title	Length	AlbumID
▶	1	Fuego	00:04:30	1
	2	Jijiji	00:06:03	2
	3	Brujeria	00:03:10	3
	4	Natural Mystic	00:03:27	4
	5	Good Times Bad Times	00:02:46	5
✱	NULL	NULL	NULL	NULL

Tabla Usuario

```
CREATE TABLE IF NOT EXISTS mellon.user (  
  UserID INT NOT NULL AUTO_INCREMENT,  
  Username VARCHAR(50) NOT NULL,  
  Email VARCHAR(255) NULL NULL,  
  Password VARCHAR(255) NOT NULL,  
  RegistrationDate DATE NULL DEFAULT NULL,  
  PRIMARY KEY (UserID),  
  UNIQUE INDEX Email (Email ASC) VISIBLE,  
  FOREIGN KEY (ArtistID)  
    REFERENCES mellon.artist (ArtistID)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);  
  
SELECT * FROM mellon.user;
```

```
INSERT INTO Album VALUES (null,'Val', 'flash@a.com', '123', null);  
INSERT INTO Album VALUES (null,'Luk', 'mood@a.com', '123', null);  
INSERT INTO Album VALUES (null,'Fran', 'rain@a.com', '123', null);  
INSERT INTO Album VALUES (null,'Num', 'cloud@a.com', '123', null);
```

	UserID	Username	Email	Password	RegistrationDate
▶	1	Val	flash@a.com	123	2024-05-14
	2	Luk	mood@a.com	123	2024-03-14
	3	Fran	rain@a.com	123	2024-08-14
	4	Num	cloud@a.com	123	2024-10-14
*	NULL	NULL	NULL	NULL	NULL

Tabla Album

```
CREATE TABLE IF NOT EXISTS mellon.album (  
  AlbumID INT NOT NULL AUTO_INCREMENT,  
  Title VARCHAR(255) NOT NULL,  
  ReleaseYear VARCHAR(100) NULL DEFAULT NULL,  
  ArtistID INT NULL DEFAULT NULL,  
  PRIMARY KEY (AlbumID),  
  INDEX (ArtistID ASC) VISIBLE,  
  FOREIGN KEY (ArtistID)  
    REFERENCES mellon.artist (ArtistID)  
    ON DELETE CASCADE  
    ON UPDATE CASCADE  
);  
  
SELECT * FROM mellon.album;  
  
INSERT INTO Album VALUES (null,'Otro dia en el planeta Tierra', '2005', null);  
INSERT INTO album VALUES (null, 'Oktubre', '2008', null);  
INSERT INTO Album VALUES (null,'Armando Camaleon', '2004', null);  
INSERT INTO Album VALUES (null,'Exodus', '1977', null);  
INSERT INTO Album VALUES (null,'Led Zeppelin', '1969', null);
```

	AlbumID	Title	ReleaseYear	ArtistID
▶	1	Otro dia en el planeta Tierra	2005	1
	2	Oktubre	2008	2
	3	Armando Camaleon	2004	3
	4	Exodus	1977	4
	5	Led Zeppelin	1969	5
*	NULL	NULL	NULL	NULL

Tabla Lista de Reproducción

```
CREATE TABLE IF NOT EXISTS mellon.playlist (  
  PlaylistID INT NOT NULL AUTO_INCREMENT,  
  Name VARCHAR(50) NOT NULL,  
  UserID INT NULL DEFAULT NULL,  
  PRIMARY KEY (PlaylistID),  
  INDEX UserID (UserID ASC) VISIBLE,  
  CONSTRAINT playlist_ibfk_1  
    FOREIGN KEY (UserID)  
    REFERENCES mellon.user (UserID),  
  CONSTRAINT artist  
    FOREIGN KEY (PlaylistID)  
    REFERENCES mellon.artist (Artist)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION,  
  CONSTRAINT song  
    FOREIGN KEY (PlaylistID)  
    REFERENCES mellon.song (SongID)  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION  
);  
  
SELECT * FROM mellon.playlist;  
  
INSERT INTO playlist VALUES (null,'Lokura', null);  
insert into playlist values (null, 'Fiesta', null);  
insert into playlist values (null, 'Dormir', null);  
INSERT INTO playlist VALUES (null,'Tranqui', null);
```

	PlaylistID	Name	UserID
▶	1	Lokura	1
	2	Fiesta	2
	3	Dormir	3
	4	Tranqui	4
✱	NULL	NULL	NULL

Insertar

```
SELECT * FROM mellon.artist;
```

```
INSERT INTO artist VALUES (null, 'Intoxicados', 'punk');
```

```
INSERT INTO artist VALUES (null, 'Patricio Rey y los redondos', 'rock');
```

```
INSERT INTO artist VALUES (null, 'Los Tipitos', 'rock');
```

```
INSERT INTO artist VALUES (null, 'Bob Marley', 'reagge');
```

```
INSERT INTO artist VALUES (null, 'Led Zeppelin', 'rock');
```

INSERT INTO artist VALUES: Inserta nuevos registros en la tabla artist.

(null, 'Intoxicados', 'punk'): Los valores que se van a insertar en cada columna de la tabla.

El resto aplica lo mismo, pero con otros datos.

Actualizar Datos

```
SELECT * FROM mellon.album;

UPDATE album
SET title = 'Otro dia en el planeta Tierra'
WHERE albumid = 1;
UPDATE album
SET title = 'Oktubre'
WHERE albumid = 2;
UPDATE album
SET title = 'Armando Camaleon'
WHERE albumid = 3;
```

“Update álbum” especifica la tabla “album” donde se realizarán las actualizaciones

SET title = 'Otro dia en el planeta Tierra': Establece el nuevo valor del campo title para el álbum con albumid igual a 1.

WHERE albumid = 1: Condición que asegura que sólo se actualizará el álbum con albumid igual a 1.

Eliminar Datos

```
DELETE FROM canciones  
WHERE artista = 'Bob Marley';
```

DELETE FROM: Instrucción que indica que se van a eliminar registros de la tabla especificada.

canciones: Nombre de la tabla de la que se eliminarán los registros.

WHERE artista = 'Bob Marley': Especifica que solo se eliminarán las canciones cuyo campo artista sea igual a 'Bob Marley'.

Consultas en MySql: WHERE

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

	AlbumID	Title	ReleaseYear	ArtistID
▶	4	Exodus	1977	4
✱	NULO	NULO	NULO	NULO

SELECT * FROM Album: Selecciona todas las columnas de la tabla Album.

WHERE ArtistID = (subconsulta): Filtra los resultados para incluir solo los registros en los que ArtistID coincide con el ArtistID devuelto por la subconsulta.

SELECT ArtistID FROM Artist: Selecciona la columna ArtistID de la tabla Artist.

WHERE Name = 'Bob Marley': Filtra los resultados para obtener solo el ArtistID del artista cuyo nombre es 'Bob Marley'.

Consultas en MySQL: WHERE between

```
SELECT * FROM user WHERE RegistrationDate BETWEEN '2024-01-01' AND '2024-12-31';
```

	UserID	Username	Email	Password	RegistrationDate
▶	1	Val	flash@a.com	123	2024-05-14
	2	Luk	mood@a.com	123	2024-03-14
	3	Fran	rain@a.com	123	2024-08-14
	4	Num	cloud@a.com	123	2024-10-14
*	NULL	NULL	NULL	NULL	NULL

SELECT *: Selecciona todas las columnas de la tabla.

FROM user: Especifica la tabla user.

WHERE RegistrationDate: Filtra los resultados basándose en la columna RegistrationDate.

BETWEEN '2024-01-01' AND '2024-12-31': Selecciona los registros cuya RegistrationDate esté entre el 1 de enero de 2024 y el 31 de diciembre de 2024.

Consultas en MySql: WHERE Limit

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

	AlbumID	Title	ReleaseYear	ArtistID
▶	1	Otro día en el planeta Tierra	2005	1
*	NULL	NULL	NULL	NULL

SELECT *: Selecciona todas las columnas de la tabla Album.

FROM Album: Especifica la tabla Album de la que se van a seleccionar los registros.

WHERE ArtistID = 1: Filtra los registros para incluir solo aquellos donde el ArtistID sea igual a 1.

LIMIT 5: Restringe el número de filas devueltas a un máximo de 5 registros.

Consultas en MySql: Inner Join

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

	SongID	Title	Length	AlbumID	AlbumID	Title	ReleaseYear	ArtistID	ArtistID	Name	Genre
▶	5	Good Times Bad Times	00:02:46	5	5	Led Zeppelin	1969	5	5	Led Zeppelin	rock

INNER JOIN album: Combina la tabla song con la tabla album utilizando el campo albumID como clave de relación.

INNER JOIN artist: Combina el resultado anterior con la tabla artist utilizando el campo artistID como clave de relación.

WHERE artist.name = 'Led Zeppelin': Filtra los resultados para incluir solo las canciones del artista cuyo nombre es 'Led Zeppelin'.

Consultas en MySql: Inner Join Con Filtros

```
SELECT song.Title AS SongTitle, album.Title AS AlbumTitle, artist.Name AS ArtistName, album.ReleaseYear  
  
FROM song  
|  
INNER JOIN album ON song.AlbumID = album.AlbumID  
  
INNER JOIN artist ON album.ArtistID = artist.ArtistID  
  
WHERE album.ReleaseYear > 2000;
```

	SongTitle	AlbumTitle	ArtistName	ReleaseYear
►	Fuego	Otro día en el planeta Tierra	Intoxicados	2005
	Jijiji	Oktubre	Patricio Rey y los redondos	2008
	Brujeria	Armando Camaleon	Los Tipitos	2004

song.Title AS SongTitle: Alias para la columna Title de la tabla song, renombrada como SongTitle.

album.Title AS AlbumTitle: Alias para la columna Title de la tabla album, renombrada como AlbumTitle.

artist.Name AS ArtistName: Alias para la columna Name de la tabla artist, renombrada como ArtistName.

album.ReleaseYear: Columna ReleaseYear de la tabla album.

INNER JOIN album ON song.AlbumID = album.AlbumID: Combina la tabla song con la tabla album utilizando el campo AlbumID como clave de relación.

INNER JOIN artist ON album.ArtistID = artist.ArtistID: Combina el resultado anterior con la tabla artist utilizando el campo ArtistID como clave de relación.

WHERE album.ReleaseYear > 2000: Filtra los resultados para incluir solo aquellos álbumes cuyo año de lanzamiento sea posterior al año 2000.

Consultas en MySql: operador Logico / comparación

```
SELECT * FROM mellon.artist;  
  
select * from artist  
      where ArtistID >= 0 and Genre = "Rock"
```

	ArtistID	Name	Genre
▶	2	Patricio Rey y los redondos	rock
	3	Los Tipitos	rock
	5	Led Zeppelin	rock
✱	NULL	NULL	NULL

SELECT * FROM mellon.artist;: Esta consulta selecciona todos los registros de la tabla artist en la base de datos mellon.

SELECT * FROM artist: Selecciona todos los registros de la tabla artist.

WHERE ArtistID >= 0 AND Genre = 'Rock': Filtra los resultados para incluir solo los registros donde:

ArtistID es mayor o igual a 0.

Genre es igual a "Rock".

Consultas en MySQL: operador Logico / comparación

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

	ArtistID	Name	Genre
▶	1	Intoxicados	punk
	4	Bob Marley	reggae
*	NULL	NULL	NULL

SELECT * FROM mellon.artist: Esta consulta selecciona todos los registros de la tabla artist en la base de datos mellon.

SELECT * FROM artist: Selecciona todos los registros de la tabla artist.

WHERE NOT Genre = 'Rock': Filtra los resultados para incluir sólo aquellos registros donde el valor de la columna Genre no sea igual a "Rock".

Consultas en MySql: Group by

```
select Genre from artist  
group by Genre
```

	Genre
▶	punk
	rock
	reggae

SELECT Genre: Selecciona la columna Genre de la tabla artist.

FROM artist: Especifica la tabla artist de la cual se van a seleccionar los datos.

GROUP BY Genre: Agrupa los resultados basándose en el valor de la columna Genre.

Consultas en MySql: Group by con función agregada

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

	Genero	Total Bandas
▶	punk	1
	rock	3
	reggae	1

SELECT Genre AS "Genero": Selecciona la columna Genre de la tabla artist y la renombra como "Genero" utilizando un alias.

COUNT(*) AS "Total Bandas": Cuenta el número total de filas para cada grupo de género y lo presenta como "Total Bandas" utilizando un alias.

FROM artist: Especifica la tabla artist de la cual se están seleccionando los datos.

GROUP BY Genre: Agrupa los resultados según el valor de la columna Genre. Esto significa que la consulta devolverá un resultado por cada género musical único presente en la tabla artist, junto con el número de bandas que pertenecen a ese género.

Gracias

