

PROYECTO BASE DE DATOS

Por Iván Naranjo Marín

Bases de datos: 1º DAW

Índice

PARTE I : DISEÑO Y CREACIÓN DEL PROYECTO	2
1- Descripción e Introducción del proyecto	2
2- Resolución del diagrama MER extendido	4
3- Modelo relacional y paso a tablas en MySQL Workbench	4
4- Conclusión	7
PARTE II: CARGA MASIVA DE DATOS Y CONSULTAS EN MYSQL	7
1- Actualización del modelo	7
2- Carga masiva de datos	8
3- Consultas	10
4- Conclusión	13

PARTE I : DISEÑO Y CREACIÓN DEL PROYECTO

1- Descripción e Introducción del proyecto

En este documento se describirán las características, los métodos y los resultados de la realización y creación del proyecto “YUI Sound”, que se basa en una empresa cuya actividad principal se basa en la organización de conciertos, como a su vez de la venta de entradas en diferentes instalaciones. Todo esto a su vez se hará a nivel nacional e internacional.

De los conciertos se quiere tener registrado tanto la fecha como la duración de dicho concierto, y de las instalaciones se tiene una ID específica, el nombre de la instalación, el país y dirección exacta donde se encuentra la instalación y la capacidad de este.

Un concierto es dependiente de una gira, la cual tiene un nombre específico.

Los conciertos disponen de entradas, estas pueden ser entradas para ver en vivo el concierto o para verlo a través de streaming; para las entradas en vivo se guarda la categoría (vip, normal, económica, etc.), la zona asignada, y un número (Ej. Si hay un concierto que dispone de 60.000 entradas, los números irán desde 00000 hasta 60000.); mientras que las entradas online solo necesitarán un código para acceder al streaming.

Las 2 entradas deberán guardar el número de entradas que dispone el concierto y el precio.

Por otra parte, del artista se deberá registrar su nombre, el país de origen, el género musical que realiza, un sueldo correspondiente y un agente/ representante como forma de contacto.

Un artista puede dar muchas giras, pero cada gira tiene registrado un artista individual.

En los conciertos del artista pueden aparecer otros colaboradores, por lo que se querrá registrar esta situación; un artista puede colaborar en muchos conciertos, y un concierto puede contar con varios artistas invitados.

Además, se quiere guardar el personal que va a trabajar en cada concierto.

Para el personal, guardaremos un identificador, nombre y apellidos, el NIF, fecha de nacimiento, el sueldo que se les pagará, el puesto que tienen en la organización y un teléfono de contacto.

Tendremos en cuenta que una persona puede ser contratada para hacer varios conciertos dentro de la empresa, y que estos conciertos además necesitan una cantidad considerable de personas.

Dados todos estos datos, podremos definir los siguientes campos con sus atributos correspondientes:

1-Instalaciones: Identificador, Nombre, País, Dirección y Capacidad.

2-Artistas: Nombre, País de origen, Sueldo, Género Musical y Agente/Representante.

3-Gira: Nombre.

4-Conciertos: Fecha del Concierto y duración. (Depende de Gira).

5-Personal: ID, Nombre, Apellidos, NIF, Fecha de nacimiento, Teléfono, Sueldo y Puesto.

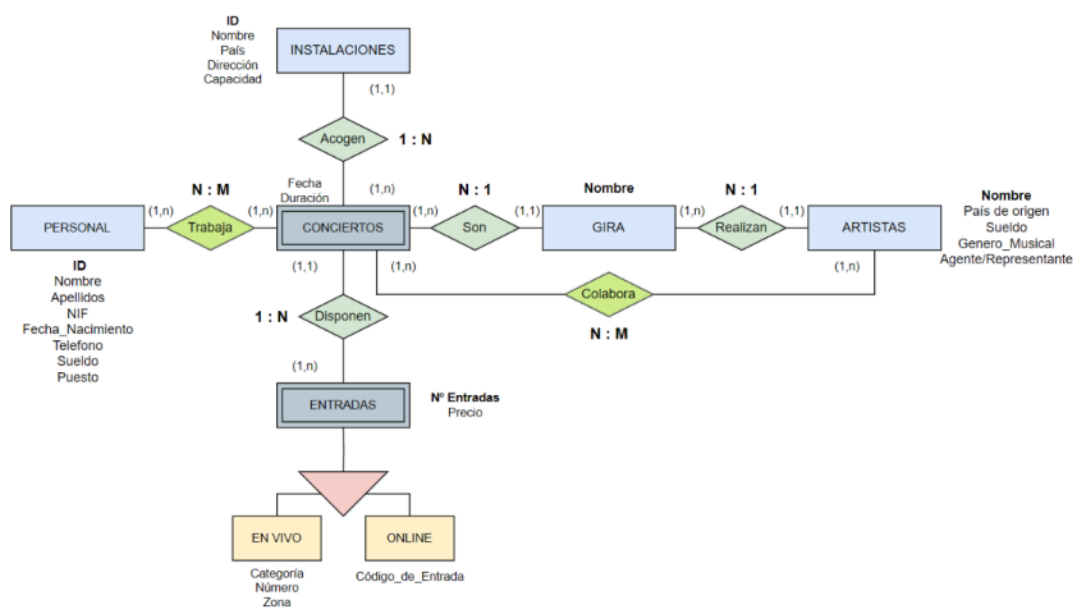
6-Entradas: N.º Entradas y Precio. (Depende de conciertos).

-Si es en vivo: Categoría, Número de entrada y Zona.

-Si es online: Código de entrada al streaming.

2- Resolución del diagrama MER extendido

Gracias a toda la información que hemos desarrollado en el ejercicio anterior, podremos diseñar un modelo Entidad/Relación como herramienta para la creación de los siguientes puntos, como el modelo relacional y la creación del Workbench posterior a este.



3- Modelo relacional y paso a tablas en MySQL Workbench

Ahora, con el modelo MER creado y diseñado, podemos empezar a crear el modelo relacional de nuestra base de datos.

Siguiendo el modelo, debemos diseñar un modelo con 8 apartados (Contando herencias y Relaciones N:M)

INSTALACIONES (Id_Instalación, Nombre, País, Dirección, Capacidad)

PK: Id_Instalación

ARTISTAS (Nombre, País de Origen, Sueldo, Género_Musical)

PK: Nombre

GIRA (Nombre, Nombre)

PK: Nombre

FK: Nombre REFERENCIA **ARTISTA**

CONCIERTOS (Fecha, Duración, Id_Instalación, Nombre)

FK: Nombre REFERENCIA **GIRA**

FK: Id_Instalación REFERENCIA **INSTALACIÓN**

ENTRADAS (Nº_Entradas, Nombre, Precio, Tipo, Categoría, Número, Zona, Código_Entradas)

PK: Nº_Entradas

FK: Nombre REFERENCIA **GIRA**

PERSONAL (Id_Personal, Nombre, Apellidos, NIF, Fecha_Nacimiento, Teléfono, Sueldo)

PK: Id_Personal

PERSONAL_TRABAJA_CONCIERTOS (Id_Personal, Nombre)

PK: Id_Personal, Nombre

FK: Id_Personal REFERENCIA **PERSONAL**

FK: Nombre REFERENCIA **GIRA**

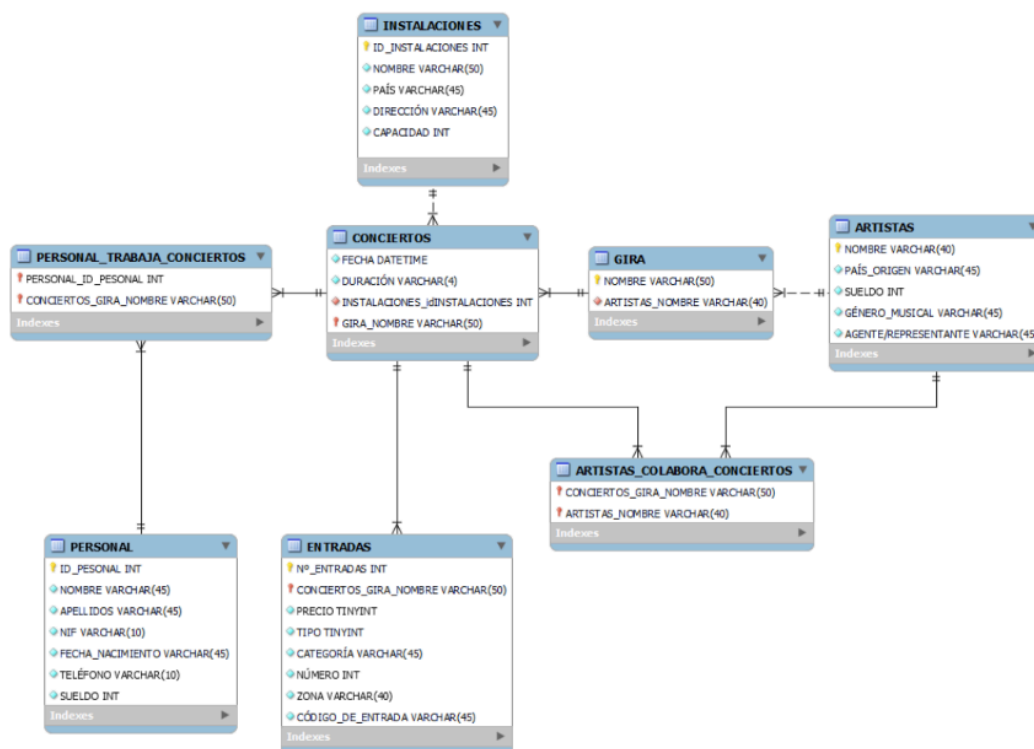
ARTISTA_COLABORA_CONCIERTOS (Nombre, Nombre)

PK: Nombre, Nombre

FK: Nombre REFERENCIA **ARTISTA**

FK: Nombre REFERENCIA **GIRA**

Gracias al modelo Entidad/Relación que hemos diseñado en el apartado anterior, y el modelo relacional realizado previamente, podemos diseñar de manera intuitiva el paso a tablas.



Dentro de MySQL podremos exportar el diseño que acabamos de crear en un script .sql para poder añadirlo a los programas para realizar consultas (En este caso el script se encuentra en Github)

<https://github.com/IvanNaranjoEducaand/PROYECTO-BASE-DE-DATOS>

4- Conclusión

Como conclusión para esta primera parte del proyecto , a nivel personal se aprende el proceso de diseño y creación de una base de datos de forma intuitiva, además de la creación de un script para poder añadir la base de datos a cualquier programa con el que se puedan realizar consultas (Como DBeaver) o de añadirle en un futuro los datos y el trato de estos de forma sencilla.

PARTE II: CARGA MASIVA DE DATOS Y CONSULTAS EN MYSQL

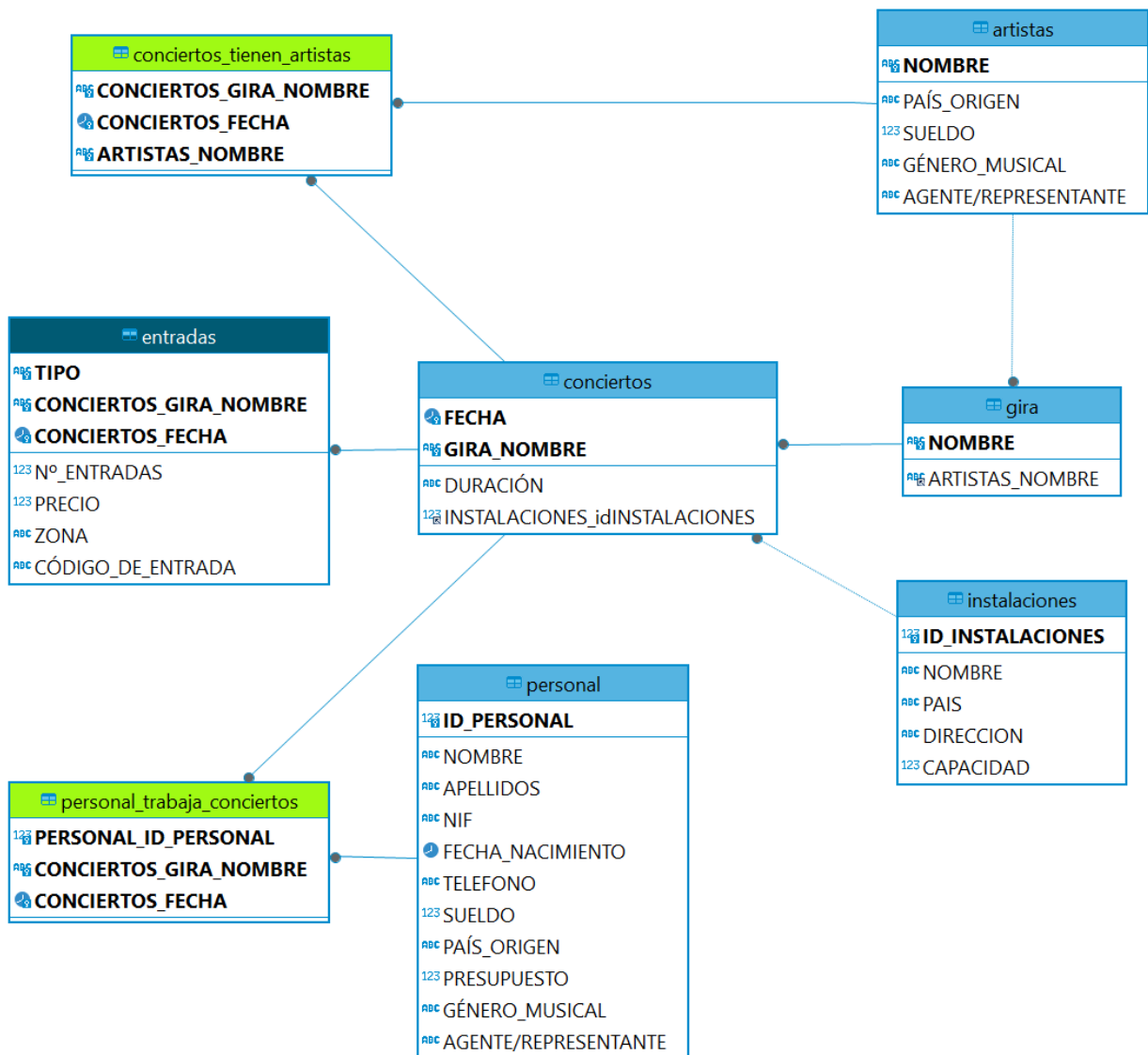
1- Actualización del modelo

Debido a las condiciones de la carga masiva de datos, se han tenido que realizar diferentes cambios en la base de datos, quedando de la siguiente forma:

Los cambios más notables son:

- La adición de fecha como clave primaria de *conciertos*
- A consecuencia del cambio anterior, las tablas de relación N:M también tienen como clave primaria las fechas de los conciertos
- Por último, la tabla *entradas* ha sido cambiada casi por completo, seleccionando el tipo de entrada, la fecha del concierto y el nombre de la gira a la que pertenece como clave primaria.

Con estos cambios ya explicados, a continuación se podrá ver el modelo relacional ya renovado en DBeaver, donde se va a realizar la carga masiva de datos.



2- Carga masiva de datos

Para este siguiente apartado, para explicar el proceso de carga de datos, colocaré el número de datos, el orden en el que las he realizado y la razón de este orden.

Para la creación de la carga masiva de datos, he utilizado tanto datos objetivos como datos inventados o generados que tuvieran sentido para las reglas de la tabla; La mayoría de datos han sido hechos a mano para que toda la información tenga sentido de forma conjunta.

1º Carga de datos: ARTISTAS, INSTALACIONES Y PERSONAL

Tanto *artistas* como *instalaciones* se comprenden de 50 filas de datos únicos respectivamente, mientras que la tabla *personal* concentra un total de 90 filas de datos.

Estas tablas se realizaron de forma inicial debido a que no tienen ningún tipo de dependencias respecto a otras tablas en el modelo.

2º Carga de datos: GIRA

La tabla *gira* cuenta con 90 registros únicos.

Esta se realizó como segunda etapa debido a la dependencia de *gira* con el nombre de los artistas que realizan estas mismas.

3º Carga de datos: CONCIERTOS

La tabla *conciertos* cuenta con 275 registros únicos por fila.

Esta carga de datos necesitaba para su realización los datos de la tabla de giras.

4º Carga de datos: ENTRADAS

La tabla *entradas* se forma con un total de 605 datos únicos, en esta tabla además se añaden las diferencias entre entradas online o entradas en vivo, que se diferencian dependiendo si el valor “Zona” o el valor “Codigo de entrada” es nulo.

Para poder crear la carga de datos masiva en esta tabla, se necesitaban los datos de la tabla conciertos.

5º Carga de datos: PERSONAL_TRABAJA_CONCIERTOS, CONCIERTOS_TIENEN_ARTISTAS

La tabla *personal_trabaja_conciertos* se compone de un total de 960 datos, mientras que la tabla *conciertos_tienen_artistas* está formada de 544 datos. Estas 2 tablas necesitaban tanto los datos de la 1º carga de datos como la 3º (*Artistas, Personal y conciertos*)

3- Consultas

Como última sección de este apartado, pasaremos a realizar unas consultas como forma de test de la base de datos y la carga masiva de datos.

1- Listado de todas las fechas y nombres de los conciertos realizados en el año 2016, ordenados de forma ascendente.

```
select c.FECHA, c.GIRA_NOMBRE
from conciertos c
where year(c.FECHA) = 2016
order by c.FECHA asc;
```

Resultado de la consulta:

	FECHA	GIRA_NOMBRE
1	2016-01-06	Formation World Tour (2016)
2	2016-01-20	A Head Full of Dreams Tour (2016-2017)
3	2016-02-09	Saint Pablo Tour (2016)
4	2016-02-14	Purpose World Tour (2016)
5	2016-03-08	Ritual Tour (2016-2017)
6	2016-03-27	Purple Reign Tour (2016)
7	2016-04-18	Anti World Tour (2016)
8	2016-04-30	Saint Pablo Tour (2016)
9	2016-05-12	Anti World Tour (2016)
10	2016-05-24	Saint Pablo Tour (2016)
11	2016-06-02	Nostalgic for the Present Tour (2016-2017)
12	2016-06-27	Saint Pablo Tour (2016)
13	2016-07-05	Summer Sixteen Tour (2016)
14	2016-07-19	Saint Pablo Tour (2016)
15	2016-08-07	Anti World Tour (2016)
16	2016-08-16	Saint Pablo Tour (2016)
17	2016-08-22	Energía Tour (2016)
18	2016-09-09	Purple Reign Tour (2016)
19	2016-09-23	Purpose World Tour (2016)
20	2016-10-12	Adele Live 2016 (2016)
21	2016-10-29	Anti World Tour (2016)
22	2016-11-03	Saint Pablo Tour (2016)
23	2016-11-15	Listen Tour (2015-2016)
24	2016-12-15	Energía Tour (2016)
25	2016-12-15	Purple Reign Tour (2016)
26	2016-12-31	Anti World Tour (2016)

2- Listado del sueldo medio por año de nacimiento, la media debe estar a redondeada 2 decimales.

```
select year(p.FECHA_NACIMIENTO) as AñoNacimiento,  
round(avg(SUELDO), 2) as mediaSalarial  
from personal p  
group by AñoNacimiento  
order by AñoNacimiento desc;
```

Resultado de la consulta:

	AñoNacimiento	mediaSalarial
1	2.008	6.318
2	2.007	4.219
3	2.006	8.006,5
4	2.005	6.028
5	2.004	4.551,8
6	2.003	5.017,33
7	2.002	7.737
8	2.001	5.164
9	2.000	6.612,33
10	1.999	7.580,67
11	1.998	7.470,5
12	1.997	8.060,33
13	1.996	5.982,33
14	1.995	4.176,33
15	1.994	6.238,67
16	1.993	5.796,67
17	1.992	5.617,33
18	1.991	7.782,75
19	1.990	6.501,5
20	1.989	4.230,67
21	1.988	6.282,5
22	1.987	6.300,33
23	1.986	5.111,67
24	1.985	5.661,33

3- Nombre de las giras del artista que tenga el mayor sueldo.

```
select *  
from gira g  
where g.ARTISTAS_NOMBRE = (  
    select a.NOMBRE  
    from artistas a  
    order by a.SUELDO desc limit 1  
);
```

Resultado de la consulta:

	NOMBRE	ARTISTAS_NOMBRE
1	24K Magic World Tour (2017-2018)	Bruno Mars
2	The Doo-Wops & Hooligans Tour (2010-2012)	Bruno Mars
3	The Moonshine Jungle Tour (2013-2014)	Bruno Mars

4- Listado de las 10 entradas más caras en la instalación con mayor capacidad.

```
select e.*
from entradas e
join conciertos c ON e.CONCIERTOS_FECHA = c.FECHA
join instalaciones i on i.ID_INSTALACIONES =
c.INSTALACIONES_idINSTALACIONES
where i.nombre = (select i2.NOMBRE
                  from instalaciones i2
                  order by i2.CAPACIDAD desc limit 1
)
order by e.PRECIO desc limit 10;
```

Resultado de la consulta:

	TIPO	Nº_ENTRADAS	PRECIO	ZONA	CÓDIGO_DE_ENTRADA	CONCIERTOS_GIRA_NOMBRE	CONCIERTOS_FECHA
1	VIP +	40.000	650	3		Born This Way Ball (2012-2013)	2013-07-19 ↗
2	VIP	200.000	545	2		Born This Way Ball (2012-2013)	2013-07-19 ↗
3	VIP	10.000	532	3		The Honeymoon Tour (2015)	2015-05-12 ↗
4	VIP	40.000	499	3		A Head Full of Dreams Tour (2016-2017)	2017-05-12 ↗
5	PREMIUM	240.000	448	3		24K Magic World Tour (2017-2018)	2018-04-30 ↗
6	NORMAL	200.000	388	1		The Honeymoon Tour (2015)	2015-05-12 ↗
7	NORMAL	200.000	357	1		A Head Full of Dreams Tour (2016-2017)	2017-05-12 ↗
8	NORMAL	240.000	300	2		El Dorado World Tour (2018-2019)	2019-12-15 ↗
9	PREMIUM	30.000	281	2		The Honeymoon Tour (2015)	2015-05-12 ↗
10	ONLINE	1.000	243		5432109876	Born This Way Ball (2012-2013)	2013-07-19 ↗

5- Seleccionar el nombre de los artistas que tengan un sueldo mayor a 150.000 o que estén registrados en el género musical 'Rap'. Realizándose con 2 consultas unidas.

```
select a.NOMBRE, a.SUELDO, a.GÉNERO_MUSICAL
from artistas a
where a.SUELDO > 150000
union
select a.NOMBRE
from artistas a
where a.GÉNERO_MUSICAL = 'Rap';
```

Resultado de la consulta:

	<small>ABC</small> NOMBRE	<small>123</small> SUELDO	<small>ABC</small> GÉNERO_MUSICAL
1	21 Savage	275.000	Rap
2	Adele	255.000	Pop
3	Ariana Grande	360.000	Pop
4	Bad Bunny	450.000	Latino
5	Beyoncé	435.000	Pop
6	Billie Eilish	240.000	Pop
7	Bruno Mars	500.000	Pop
8	Calvin Harris	360.000	Electronica
9	Coldplay	460.000	Pop
10	David Guetta	200.000	Electronica
11	Drake	500.000	Rap
12	Dua Lipa	450.000	Pop
13	Ed Sheeran	450.000	Pop
14	Elton John	325.000	Pop
15	Eminem	450.000	Rap
16	Harry Styles	500.000	Pop
17	Imagine Dragons	450.000	Pop
18	Justin Bieber	250.000	Pop
19	Kanye West	500.000	Rap
20	Kelly Clarkson	320.000	Pop
21	Kendrick Lamar	250.000	Rap
22	Lady Gaga	500.000	Pop
23	Lana Del Rey	250.000	Pop
24	Mariah Carey	320.000	Pop
25	Maroon 5	360.000	Pop
26	Marshmello	200.000	Electronica

4- Conclusión

Para finalizar esta parte del proyecto, y como en la parte anterior, en esta fase del proyecto he aprendido a nivel personal la forma de actualización del diseño inicial para que todo funcione correctamente a la hora de cargar datos, la propia investigación para la carga masiva de datos, y finalmente, el proceso de realizar y diseñar las consulta en MySQL.