



Universidad Autónoma de Baja California Sur

BASE DE DATOS AVANZADAS

PRÁCTICA #1

Alumno: Briones Herrera Jazziel Abdiel

Carrera:

Ingeniería en Desarrollo de Software

Maestro:

Mtra. Irma Leticia Palacios López

Fecha de entrega: 24/02/2023



El primer paso en todas las prácticas será ejecutar la sentencia “show tables;” y pegar una captura de pantalla de la sentencia. Así como la sentencia “describe <nombreTabla>” y la captura de cada una. **Ojo. Este último paso se repetirá según la cantidad de tablas que existen en la base de datos.**

show tables;

108 • `show tables;`

Result Grid | Filter Rows:

Tables_in_musica
album
artista
cancion
contenido_album
disquera
genero
idioma

Result 1 x

```
8
9  #CREACION DE TABLA DISQUERA ---PASO 2
10 • describe DISQUERA;
11 • create table DISQUERA(
12     id_disquera int auto_increment,
13     constraint pk_id_disquera primary key (id_disquera),
14     nombre varchar(50) not null,
15     sede varchar(20) default "DESCONOCIDO"
16 );
17
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: ☐

Field	Type	Null	Key	Default	Extra
id_disquera	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	NO		NULL	
sede	varchar(20)	YES		DESCONOCIDO	



DASC

```
17
18 #CREACION DE TABLA GENERO ---PASO 3
19 • describe GENERO;
20 • create table GENERO(
21     id_genero int auto_increment,
22     constraint pk_id_genero primary key (id_genero),
23     nombre varchar(50),
24     constraint genero01 unique (nombre)
25 );
```

Field	Type	Null	Key	Default	Extra
id_genero	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	YES	UNI	NULL	

```
34 #CREACION DE TABLA IDIOMA --PASO 6
35 • describe IDIOMA;
36 • create table IDIOMA(
37     id_idioma int auto_increment,
38     constraint pk_id_idioma primary key(id_idioma),
39     nombre varchar(50),
40     constraint Unq_nombre unique(nombre)
41 );
42
```

Field	Type	Null	Key	Default	Extra
id_idioma	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	YES	UNI	NULL	

```
43 #CREAR TABLA ARTISTA ---PASO 7
44 • describe ARTISTA;
45 • create table ARTISTA(
46     id_artista int auto_increment,
47     constraint pk_id_artista primary key(id_artista),
48     nombre varchar(50) not null,
49     nacionalidad varchar(50) default "DESCONOCIDA",
50     fecha_nac date
51 );
```

Field	Type	Null	Key	Default	Extra
id_artista	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	NO		NULL	
nacionalidad	varchar(50)	YES		DESCONOCIDA	
fecha_nac	date	YES		NULL	



DASC

```
52
53 #CREAR TABLA ALBUM ---PASO 8
54 • describe ALBUM;
55 • create table ALBUM(
56     id_album int auto_increment,
57     constraint pk_id_album primary key(id_album),
58     nombre varchar(50) not null,
59     fecha_publicacion date,
60     id_disquera int,
61     constraint fk_id_disquera foreign key(id_disquera) references DISQUERA(id_disquera),
62     precio int,
63     constraint chk_precio check(precio > 0),
64     id_artista int,
65     constraint fk_id_artista foreign key(id_artista) references ARTISTA(id_artista)
66 );
67
```

result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Field	Type	Null	Key	Default	Extra
id_album	int	NO	PRI	NULL	auto_increment
nombre	varchar(50)	NO		NULL	
fecha_publicacion	date	YES		NULL	
id_disquera	int	YES	MUL	NULL	
precio	int	YES		NULL	
id_artista	int	YES	MUL	NULL	
en_existencia	int	YES		0	

```
--
69 #CREAR TABLA CANCION --PASO 9
70 • describe CANCION;
71 • create table CANCION(
72     id_cancion int auto_increment,
73     constraint pk_id_cancion primary key (id_cancion),
74     titulo varchar(50) not null,
75     id_idioma int,
76     constraint fk_id_idioma foreign key (id_idioma) references IDIOMA(id_idioma),
77     id_genero int,
78     constraint fk_id_genero foreign key (id_genero) references GENERO(id_genero)
79 );
80
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

Field	Type	Null	Key	Default	Extra
id_cancion	int	NO	PRI	NULL	auto_increment
titulo	varchar(50)	NO		NULL	
id_idioma	int	YES	MUL	NULL	
id_genero	int	YES	MUL	NULL	



DASC

```
97  /*La siguiente tabla es CONTENIDO_ALBUM, contiene solo dos columnas en
98  donde tendremos dos claves foráneas y una clave primaria compuesta. Primero
99  se crearán la columna ID_ALBUM y la columna ID_CANCION, después
100  crearemos una restricción de clave primaria haciendo referencia a ID_ALBUM y
101  ID_CANCION. A continuación, crearemos una restricción de clave foránea con
102  ID_ALBUM haciendo referencia a la tabla ALBUM, y una restricción clave
103  foránea con ID_CANCION haciendo referencia a la tabla de CANCION, ---PASO 12*/
104  • describe CONTENIDO_ALBUM;
105  • create table CONTENIDO_ALBUM(
106      id_album int,
107      constraint fk_id_album foreign key (id_album) references ALBUM(id_album),
108      id_cancion int,
109      constraint fk_id_cancion foreign key(id_cancion) references CANCION(id_cancion),
110      constraint pk_id_album_cancion primary key (id_album, id_cancion)
111  );
112
113
```

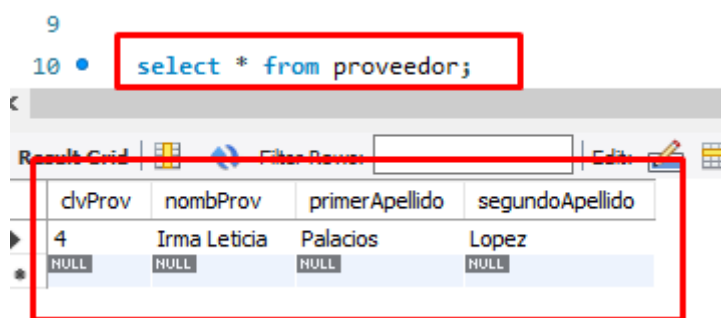
Result Grid						
Filter Rows:		Export:		Wrap Cell Content:		
	Field	Type	Null	Key	Default	Extra
▶	id_album	int	NO	PRI	NULL	
	id_cancion	int	NO	PRI	NULL	



NOTA: esto no se agrega aún, solo creación de tablas.

Para cada sentencia **SELECT**. Se hará lo mismo. Se debe describir qué es lo que mostrará la sentencia, seguido de la sentencia y la captura de pantalla de los resultados que arroja dicha sentencia.

- ***Muestra los datos de los proveedores.***



Es importante que se vea cual es la sentencia y cual es el resultado de la sentencia.

También es válido mostrar separados sentencia y captura. Ejemplo:

`select * from proveedor;`

dvProv	nombProv	primerApellido	segundoApellido
4	Irma Leticia	Palacios	Lopez
NULL	NULL	NULL	NULL