



# 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: 27/02/2023



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	

```
136 • describe CANCION;
137 • create table CANCION(
138     id_cancion int auto_increment,
139     constraint pk_id_cancion primary key (id_cancion),
140     titulo varchar(50) not null,
141     id_idioma int,
142     constraint fk_id_idioma foreign key (id_idioma) references IDIOMA(id_idioma),
143     id_genero int,
144     constraint fk_id_genero foreign key (id_genero) references GENERO(id_genero),
145     id_album int,
146     constraint fk_id_album foreign key (id_album) references ALBUM(id_album)
147
148 );
```

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	
id_album	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		



DASC

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.**

```
313 • select * from album;
```

	id_album	nombre	fecha_publicacion	id_disquera	precio	id_artista	en_existencia
▶	1	Thriller	1982-11-29	1	448	1	10
	2	Beat it	2006-03-13	1	448	1	30
	3	Bad	1987-08-31	1	144	1	50
	4	Billie Jean	1982-02-02	1	69	1	20
	5	Billie Jean	1982-02-02	1	69	1	40
	6	TIBURON	2020-02-02	1	69	2	10
	7	AMANE CER	2021-02-02	1	209	2	30
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
314
```

```
315 • select * from artista;
```

```
316
```

	id_artista	nombre	nacionalidad	fecha_nac
▶	1	Michael Jackson	EUU	1958-08-26
	2	BAD BUNNY	DESCONOCIDA	1994-03-10
*	NULL	NULL	NULL	NULL

```
316 • select * from cancion;
```

	id_cancion	titulo	id_idioma	id_genero	id_album
▶	1	Billi jean	1	1	1
	2	Amanecer	2	2	7
	3	Heal the World	1	1	3
	4	Dirty diana	1	1	4
	5	Earth song	1	1	5
	6	Dangerous	1	1	3
	7	P.Y.T	1	1	2
	8	Human nature	1	1	3
	9	Jam	1	1	3
	10	Who is it	1	1	3
	11	Tiburón	2	2	6
	12	OJOS LINDOS	2	2	7
*	NULL	NULL	NULL	NULL	NULL



DASC

323 • `select * from contenido_album;`

324

Result Grid   Filter Rows:

	id_album	id_cancion
	4	4
	4	5
	5	5
	5	6
	6	6
	7	7
	7	8
*	NULL	NULL

326 • `select * from disquera;`

327

Result Grid   Filter Rows:

Edit: 

	id_disquera	nombre	sede
▶	1	Motown	Nueva York
	2	SONY	Minato, Tokio, Japón
	3	EMI	Londres Inglaterra
	4	UNIVERSAL	Paris
*	NULL	NULL	NULL

325 • `select * from genero;`

326

Result Grid   Filter Rows:

	id_genero	nombre
▶	4	CLASICA
	3	JAZZ
	1	pop
	2	REGUETON
*	NULL	NULL

326 • `select * from idioma;`

327

Result Grid   Filter Rows:

	id_idioma	nombre
▶	2	ESPAÑOL
	1	INGLES
*	NULL	NULL



## Part1

```
/* Muestra la información de todas las canciones mostrando el título, idioma, género,
nombre del álbum y el nombre del artista.
*/
CREATE VIEW INFO_CANCION as
SELECT CANCION.titulo AS CANCION, IDIOMA.nombre AS IDIOMA, GENERO.nombre AS GENERO, ALBUM.nombre AS ALBUM, ARTISTA.nombre AS ARTISTA
FROM CANCION
JOIN album ON cancion.id_album = album.id_album
JOIN artista ON album.id_artista = artista.id_artista
JOIN idioma ON cancion.id_idioma = idioma.id_idioma
JOIN genero ON cancion.id_genero = genero.id_genero ;
```

## Part2

```
262 • select * from INFO_CANCION;
```

Result Grid					
Filter Rows:					
Export: Wrap Cell Co					
	CANCION	IDIOMA	GENERO	ALBUM	ARTISTA
▶	Who is it	INGLES	pop	Bad	Michael Jackson
	Jam	INGLES	pop	Bad	Michael Jackson
	Human nature	INGLES	pop	Bad	Michael Jackson
	P.Y.T	INGLES	pop	Beat it	Michael Jackson
	Dangerous	INGLES	pop	Bad	Michael Jackson
	Earth song	INGLES	pop	Billie Jean	Michael Jackson
	Dirty diana	INGLES	pop	Billie Jean	Michael Jackson
	Heal the World	INGLES	pop	Bad	Michael Jackson
	Billi jean	INGLES	pop	Thriller	Michael Jackson
	OJOS LINDOS	ESPAÑOL	REGUETON	AMANECER	BAD BUNNY
	Tiburón	ESPAÑOL	REGUETON	TIBURON	BAD BUNNY
	Amanecer	ESPAÑOL	REGUETON	AMANECER	BAD BUNNY





# DASC

```
265 /*Muestra el nombre del álbum, la fecha de publicación y la cantidad de canciones
266 que tiene.*/
267 • select ALBUM.nombre as "NOMBRE", ALBUM.fecha_publicacion as "FECHA PUBLICADA", ALBUM.en_existencia as "EXISTENTE"
268 from ALBUM;
```

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

	NOMBRE	FECHA PUBLICADA	EXISTENTE
▶	Thriller	1982-11-29	10
	Beat it	2006-03-13	30
	Bad	1987-08-31	50
	Billie Jean	1982-02-02	20
	Billie Jean	1982-02-02	40
	TIBURON	2020-02-02	10
	AMANECEER	2021-02-02	30

```
270 /*Muestra el nombre del artista y el total de álbumes que tiene*/
271 • SELECT ARTISTA.NOMBRE AS 'NOMBRE DEL ARTISTA', COUNT(*) AS 'TOTAL DE ÁLBUMES'
272 FROM ARTISTA
273 JOIN ALBUM ON ARTISTA.ID_ARTISTA = ALBUM.ID_ARTISTA
274 GROUP BY ARTISTA.NOMBRE;
```

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

	NOMBRE DEL ARTISTA	TOTAL DE ÁLBUMES
▶	Michael Jackson	5
	BAD BUNNY	2

```
276 /* Muestra la información del álbum más caro.*/
277 • SELECT id_album AS 'ID ALBUM', nombre AS 'NOMBRE', fecha_publicacion AS 'FECHA PUBLICACIÓN',
278 id_disquera AS 'ID DISQUERA', PRECIO AS 'PRECIO', id_artista AS 'ID ARTISTA', en_existencia AS 'EN EXISTENCIA'
279 FROM ALBUM
280 WHERE precio = (
281 SELECT MAX(PRECIO)
282 FROM ALBUM
283 );
```

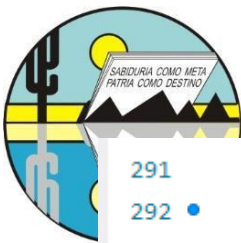
Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

	ID ALBUM	NOMBRE	FECHA PUBLICACIÓN	ID DISQUERA	PRECIO	ID ARTISTA	EN EXISTENCIA
▶	1	Thriller	1982-11-29	1	448	1	10
	2	Beat it	2006-03-13	1	448	1	30

```
285 /*Muestra los idiomas que hay y cuántas canciones hay de ese idioma.*/
286 • select i.nombre as "IDIOMA", COUNT(c.id_cancion) AS "NUMERO DE CANCIONES"
287 FROM IDIOMA i
288 LEFT JOIN CANCION c ON c.id_idioma = i.id_idioma
289 GROUP BY i.nombre;
290
```

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

	IDIOMA	NUMERO DE CANCIONES
▶	ESPAÑOL	3
	INGLES	9



RS SC

```
291 /*Muestra el género que hay y cuántas canciones hay de ese género.*/  
292 • SELECT g.nombre AS 'GENERO', COUNT(*) AS 'CANCIONES'  
293 FROM CANCION c  
294 JOIN GENERO g ON c.id_genero = g.id_genero  
295 GROUP BY g.id_genero;  
296
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
GENERO	CANCIONES		
pop	9		
REGUETON	3		

```
297 /*Muestra los artistas, su nacionalidad y cual es el promedio del precio del álbum de  
298 cada artista.*/  
299 • SELECT a.nombre AS ARTISTA, a.nacionalidad, AVG(b.precio) AS PROMEDIO_PRECIO  
300 FROM ARTISTA a  
301 JOIN ALBUM b ON a.id_artista = b.id_artista  
302 GROUP BY a.nombre, a.nacionalidad;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
ARTISTA	nacionalidad	PROMEDIO_PRECIO	
Michael Jackson	EUU	235.6000	
BAD BUNNY	DESCONOCIDA	139.0000	

```
304 /*Muestra los datos del artista, nombre, nacionalidad, fecha de nacimiento y calcula  
305 la edad que tiene el artista hasta el día de hoy usando tu función creada en el  
306 punto 7.*/  
307 • SELECT nombre, nacionalidad, fecha_nac AS 'FECHA NACIMIENTO', CUANTOS_AÑOS(fecha_nac)  
308 AS 'EDAD ACTUAL' FROM ARTISTA;  
309
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
nombre	nacionalidad	FECHA NACIMIENTO	EDAD ACTUAL
Michael Jackson	EUU	1958-08-26	64
BAD BUNNY	DESCONOCIDA	1994-03-10	28



```
310 /*Muestra la información de las disqueras y ordenalo según el nombre de su sede*/  
311 • SELECT id_disquera AS 'ID DISQUERA', nombre AS nombre, sede FROM DISQUERA ORDER BY sede;  
312
```

Result Grid			
Filter Rows: <input type="text"/>			
Export: <input type="button" value="Export"/>			
Wrap Cell Content: <input type="button" value="Wrap"/>			
	ID DISQUERA	nombre	sede
▶	3	EMI	Londres inglaterra
	2	SONY	Minato, Tokio, Japón
	1	Motown	Nueva York
	4	UNIVERSAL	Paris

Nota: en la tabla cancion se modifiko y se le agrego id\_album ya que en el pdf se le pide añadir en el procedure id\_album y otros errores que no me acuerdo xd