

# Examen Base de Datos I

dcr7juilland@gmail.com [Cambiar de cuenta](#)



Borrador guardado

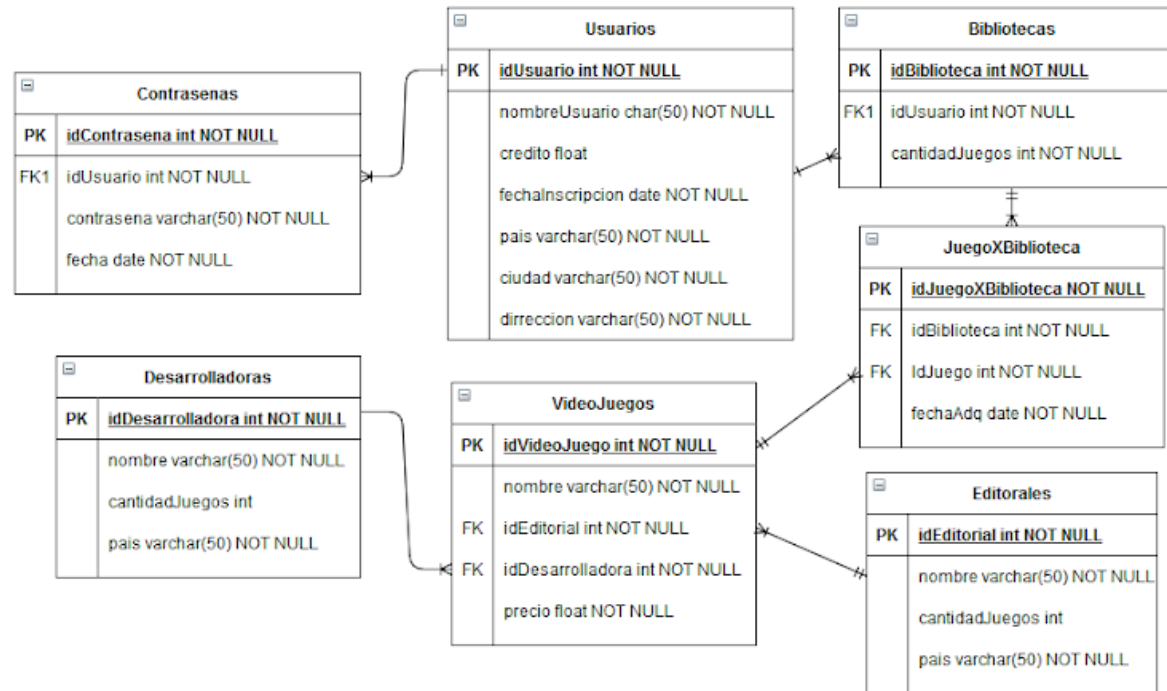
**\*Obligatorio**

Preguntas



Basándonos en el siguiente DER. Esta query retorna un listado de los usuarios cuyo ID esté entre 1 y 100 y que su nombre de usuario comience con A: \*

```
SELECT * FROM usuarios WHERE idUsuario BETWEEN 1 AND 100 AND  
nombreUsuario LIKE "A%";
```



☐ VERDADERO

☒ FALSO



La siguiente query



```
SELECT country.Name, COALESCE(country.Population, city.Population)
FROM country
LEFT JOIN city ON country.Code = city.CountryCode
WHERE country.Population IS NOT NULL
```

- ☒ Siempre muestra la población de los países
- ☐ Siempre muestra la población de las ciudades
- ☐ A veces muestra la población de los países y a veces la de las ciudades, dependiendo de los datos de la base de datos.



La ejecución de consulta del siguiente gráfico, ¿Qué datos devuelve? \*



```

SELECT cl.id, cl.color FROM coche ch
INNER JOIN marca ma ON ma.id = ch.id_marca
RIGHT JOIN color cl ON cl.id = ch.id_color
WHERE ch.id_color IS NULL;

```

coche					
<i>id</i>	<i>d_marca</i>	<i>modelo</i>	<i>anio</i>	<i>d_color</i>	<i>precio</i>
1	3	UNO	2000	1	\$ 250.000,00
2	4	FIESTA MA	2015	3	\$ 514.000,00
3	3	SIENA	2008	2	\$ 288.000,00
4	3	GRAN SIEN	2020	1	\$ 1.720.400,00
5	4	FALCON	1980	4	\$ 170.000,00
6	3	UNO	2009	2	\$ 385.000,00
7	8	COROLLA	2018	1	\$ 2.250.500,00
8	1	CORSA	2012	1	\$ 413.500,00
9	7	SANDERO	2016	4	\$ 937.500,00
10	4	FIESTA MA	2012	3	\$ 394.800,00

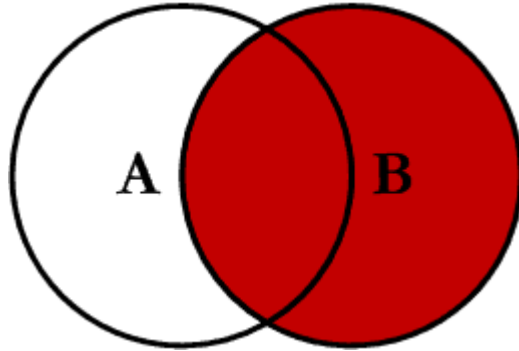
color	
<i>id</i>	<i>color</i>
1	ROJO
2	AZUL
3	VERDE
4	CELESTE
5	BLANCO
6	GRIS
7	NEGRO

marca	
<i>id</i>	<i>marca</i>
1	CHEVROL
2	DODGE
3	FIAT
4	FORD
6	NISSAN
7	RENAULT
8	TOYOTA

⦿ Devuelve el id y color de coches que no existen en la base de datos.

- ☐ Devuelve el id y color de coches que existen en la base de datos.
- ☐ La ejecución de la consulta no retorna datos.

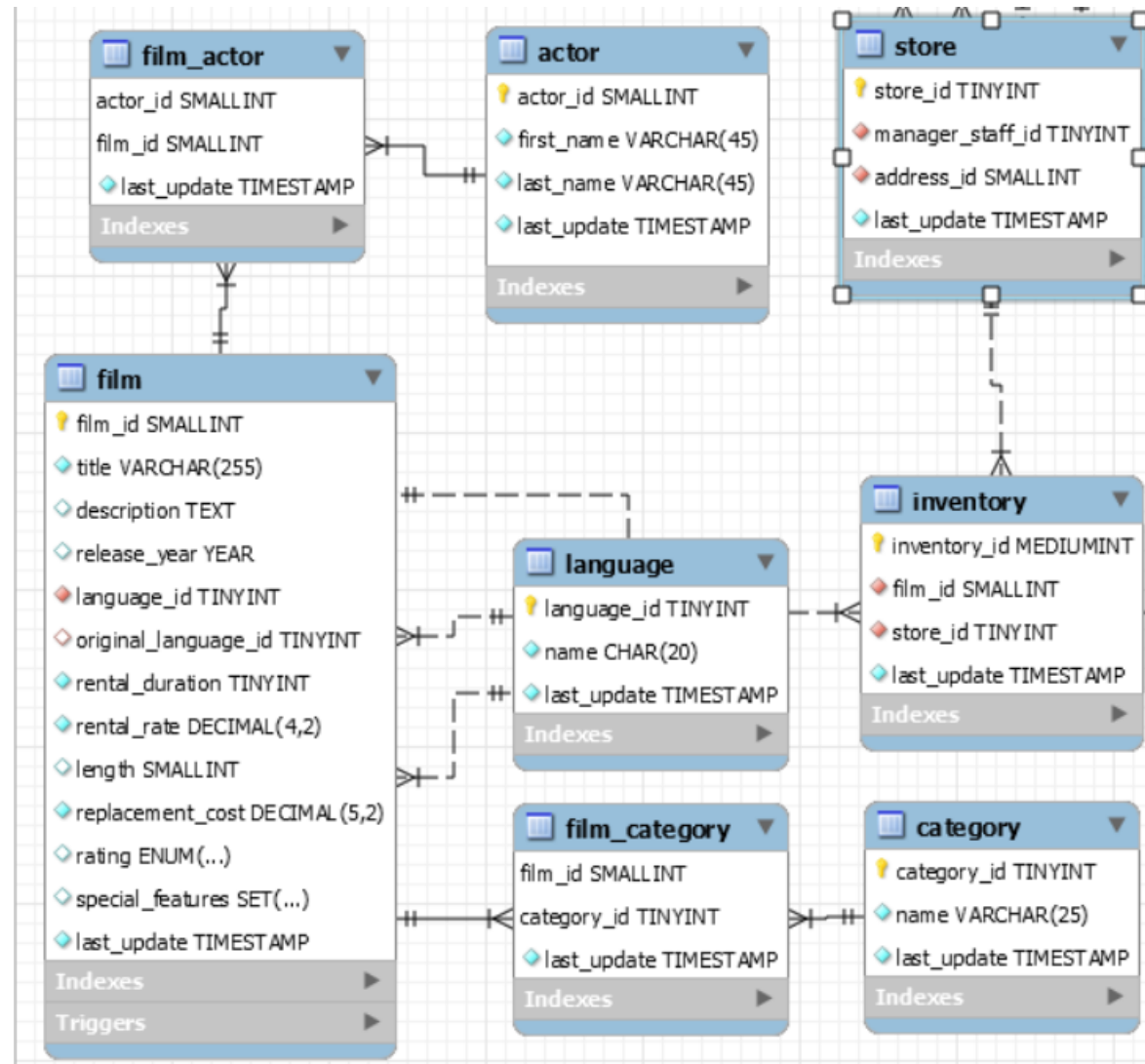
La siguiente imagen representa a \*



- ☐ INNER JOIN
- ☐ LEFT JOIN
- ☒ RIGHT JOIN
- ☐ GROUP BY



Realizar un reporte de Actores, cuyas películas contengan la categoría "children" \*



○ SELECT film.title, film\_category.category\_id, [category.name](#) FROM actor INNER JOIN film\_actor ON actor.actor\_id = film\_actor.actor\_id INNER JOIN film ON film.film\_id = film\_actor.film\_id INNER JOIN film\_category ON film.film\_id = film\_category.film\_id WHERE [category.name](#) = "children";



- ☐ SELECT actor.first\_name, film.title, film\_category.category\_id, [category.name](#) FROM actor LEFT JOIN film\_actor ON film\_actor.actor\_id = film\_actor.actor\_id LEFT JOIN film ON film.film\_id = film.film\_id LEFT JOIN film\_category ON film\_category.film\_id = film\_category.film\_id LEFT JOIN category ON category.category\_id = category.category\_id WHERE [category.name](#) = "children";
- ☒ SELECT actor.first\_name, film.title, film\_category.category\_id, [category.name](#) FROM actor INNER JOIN film\_actor ON actor.actor\_id = film\_actor.actor\_id INNER JOIN film ON film.film\_id = film\_actor.film\_id INNER JOIN film\_category ON film.film\_id = film\_category.film\_id INNER JOIN category ON film\_category.category\_id = category.category\_id WHERE [category.name](#) = "children";





Observe la consulta del siguiente gráfico y seleccione la respuesta correcta \*

**SELECT \* FROM cliente JOIN telefono;**

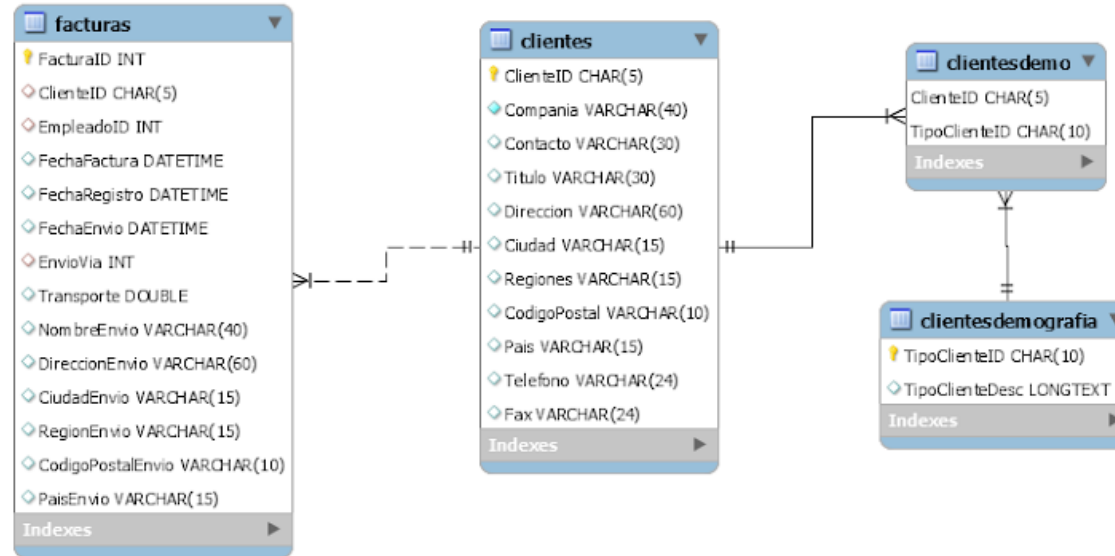
cliente			
<i>id</i>	<i>apellido</i>	<i>nombre</i>	<i>correo</i>
1	AGUILERA	ESTEBAN	aguilera80@gmail.com
2	AGUIRRE	LUCIANA	aguirre.sj@gmail.com
3	FLORES	ROBERTO	flores2000@gmail.com
4	MANRIQUE	MARCELA	manrique.mar@gmail.com
5	PEREZ	ALEJANDRO	sur25perez@gmail.com

telefono			
<i>id</i>	<i>id_cliente</i>	<i>prefijo</i>	<i>numero</i>
1	3	0264	4245588
2	1	0261	4218877
3	2	0351	43355141
4	2	011	43377125
5	1	0264	4214277
6	5	0261	4285500
7	1	011	43281214

- ☐ La consulta no se ejecutará porque tiene un error de sintaxis.
- ☒ La consulta no se ejecutará porque tiene un error en la semántica del lenguaje.
- ☐ La sintaxis de la consulta es correcta y se ejecutará satisfactoriamente.
- ☐ Ninguna respuesta es correcta.



¿Cuál de las siguientes sentencias SQL nos muestra los datos de las facturas y de los clientes cuando la ciudad de envío es Berlin? \*



- ☐ SELECT \* FROM emarket.facturas f INNER JOIN clientes c ON f.ClienteID = c.ClienteID WHERE CiudadEnvio LIKE "Berlin";
- ☐ SELECT \* FROM emarket.facturas f INNER JOIN clientes c ON f.ClienteID = c.ClienteID LIKE "Berlin";
- ☐ SELECT \* FROM emarket.facturas f INNER JOIN clientes c ON f.Cliente = "Berlin";
- ☒ Ninguna de las anteriores

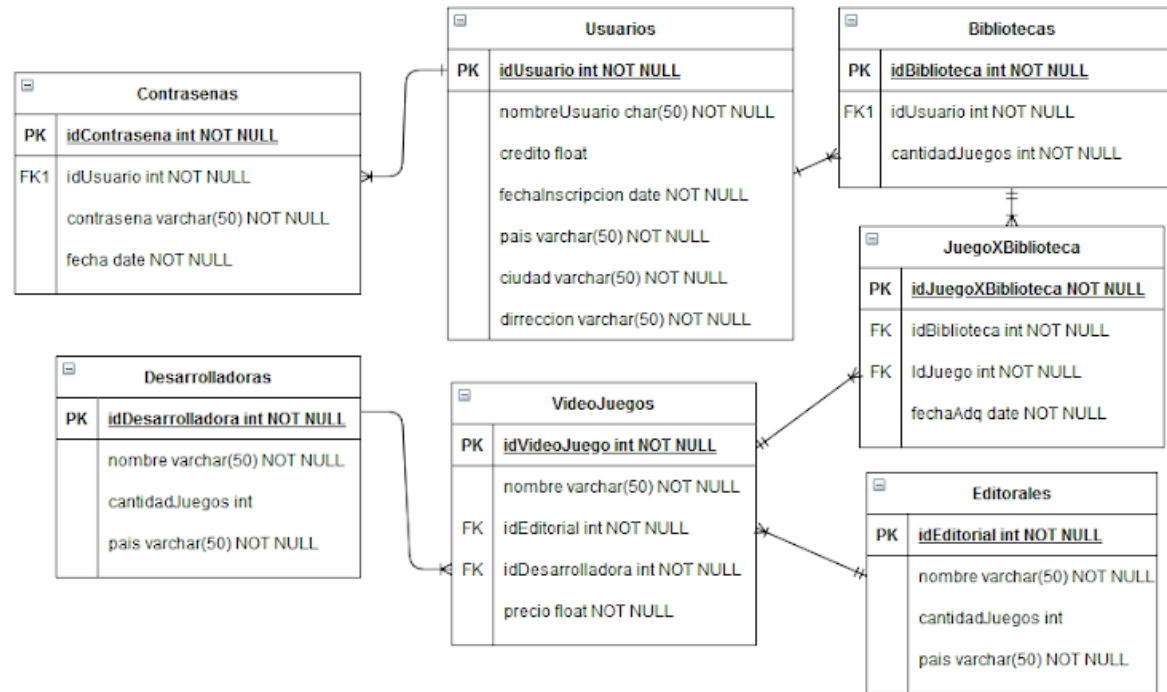


La siguiente consulta: `SELECT nombre_pelicula, genero_pelicula ORDER BY nombre_pelicula WHERE nombre_pelicula LIKE 'Harry Potter'` \*

- ☐ Devuelve todas las películas de Harry Potter ordenadas alfabéticamente por nombre
- ☐ Da error porque al LIKE le falta el signo '%'
- ☐ Devuelve la película llamada 'Harry Potter'.
- ☒ Da error de sintaxis.



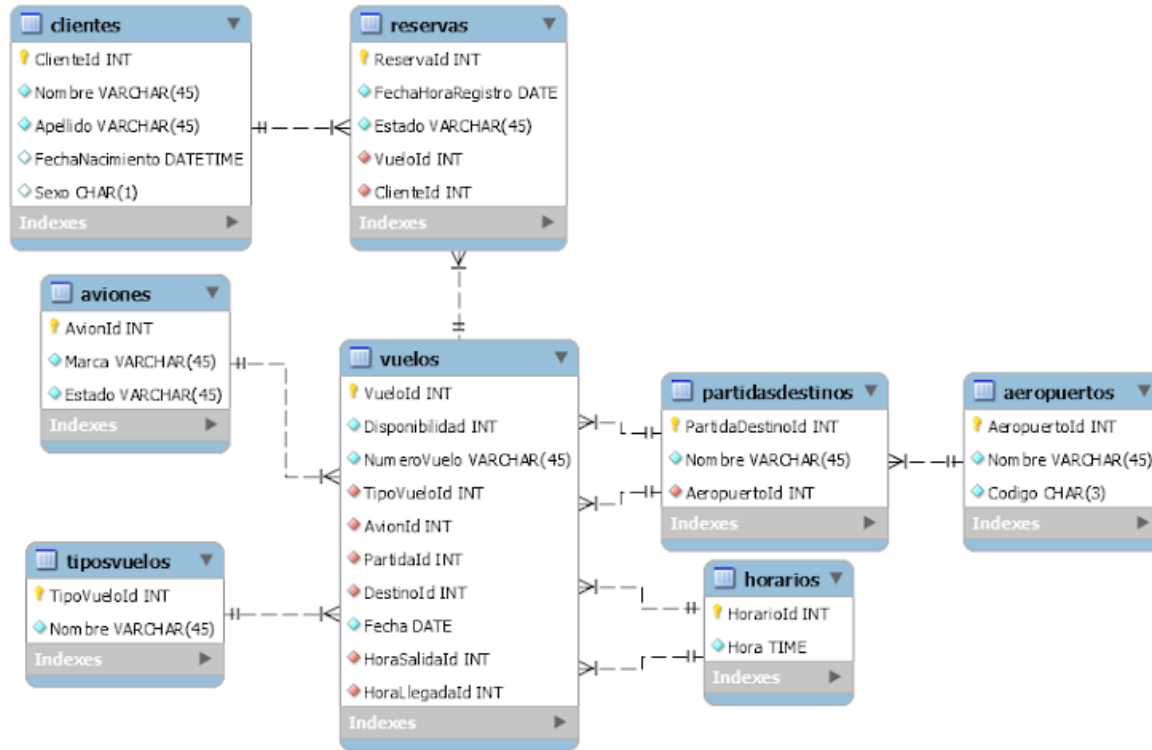
Basándonos en el siguiente DER. Si queremos obtener un listado con los países y \* la cantidad de usuarios que tienen la query sería:



- ☒ SELECT pais, COUNT(idUsuario) FROM usuarios GROUP BY idUsuario;
- ☐ SELECT pais, SUM(pais) FROM usuarios GROUP BY pais;
- ☐ SELECT pais, COUNT(idUsuario) FROM usuarios GROUP BY pais;
- ☐ SELECT pais, SUM(idUsuario) FROM usuarios GROUP BY pais;



¿Cuál de las siguientes sentencias de SQL lista en orden alfabético (sin repetir) las marcas de los aviones que tienen vuelo en la fecha 3/10/21? \*



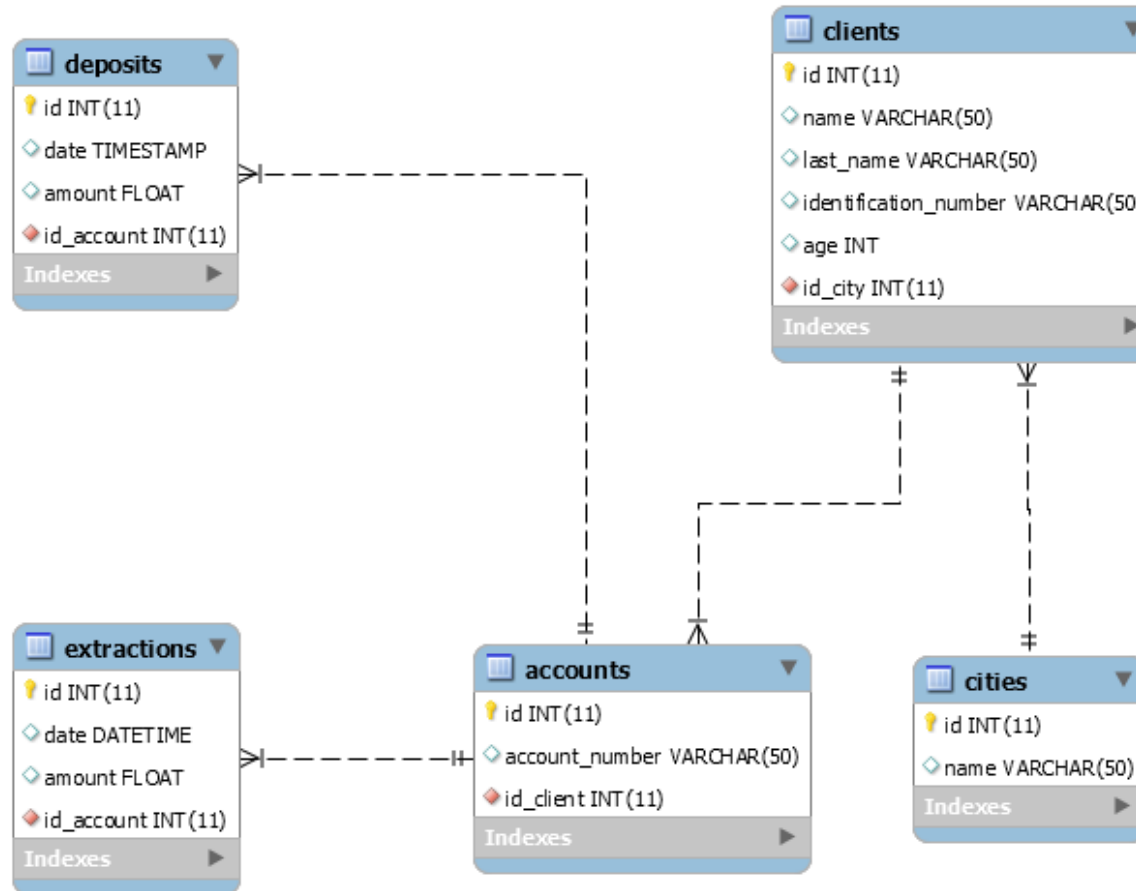
- ☐ SELECT \* FROM aviones INNER JOIN vuelos ON aviones.AvionId = vuelos.VueloId WHERE Fecha = '2021-10-03' ORDER BY Marca
- ☐ SELECT DISTINCT Marca FROM aviones INNER JOIN vuelos ON aviones.AvionId = vuelos.VueloId ORDER BY Marca
- ☐ SELECT DISTINCT Marca FROM aviones INNER JOIN vuelos ON aviones.AvionId = vuelos.VueloId WHERE Fecha = '2021-10-03'
- ☒ Ninguna de las anteriores

Una agenda telefónica en Excel, ¿Es un SGBD? \*

- ☐ No, Excel no es un SGBD
- ☐ Sí, Excel es considerado como una base de datos.
- ☒ No, una base de datos debe contener relaciones
- ☐ Si, es una base de datos SQL



¿Cuál de las siguientes sentencias de SQL muestra el nombre y apellido del cliente y la cantidad de cuentas que tiene asociado a él, ordenado de menor a mayor? \*



- ☒ SELECT [c.name](#), c.last\_name, COUNT([a.id](#)) as cant FROM clients c INNER JOIN accounts a ON [c.id](#) = a.id\_client GROUP BY [c.name](#), c.last\_name ORDER BY cant
- ☐ SELECT c.last\_name, COUNT([a.id](#)) as cant FROM clients c INNER JOIN accounts a ON [c.id](#) = a.id\_client GROUP BY [c.name](#), c.last\_name ORDER BY cant



- ☐ SELECT [c.name](#), COUNT([a.id](#)) as cant FROM clients c INNER JOIN accounts a ON [c.id](#) = a.id\_client GROUP BY [c.name](#), c.last\_name ORDER BY cant





La ejecución de la consulta del siguiente gráfico, ¿Devuelve uno o más registros con algún dato nulo? \*

```
SELECT * FROM cliente  
RIGHT JOIN telefono ON telefono.id_cliente = cliente.id;
```

cliente			
id	apellido	nombre	correo
1	AGUILERA	ESTEBAN	aguilera80@gmail.com
2	AGUIRRE	LUCIANA	aguirre.sj@gmail.com
3	FLORES	ROBERTO	flores2000@gmail.com
4	MANRIQUE	MARCELA	manrique.mar@gmail.com
5	PEREZ	ALEJANDRO	sur25perez@gmail.com

telefono			
id	id_cliente	prefijo	numero
1	3	0264	4245588
2	1	0261	4218877
3	2	0351	43355141
4	2	011	43377125
5	1	0264	4214277
6	5	0261	4285500
7	1	011	43281214

- ☐ Si, devuelve un registro con datos parcialmente nulos.
- ☐ Si, devuelve cinco registros con datos parcialmente nulos.
- ☒ Si, devuelve siete registros con datos parcialmente nulos.



☐ No devuelve registros con datos parcialmente nulos.



Observe la consulta del siguiente gráfico y seleccione la respuesta correcta \*

**SELECT \* FROM cliente JOIN telefono;**

cliente			
<i>id</i>	<i>apellido</i>	<i>nombre</i>	<i>correo</i>
1	AGUILERA	ESTEBAN	aguilera80@gmail.com
2	AGUIRRE	LUCIANA	aguirre.sj@gmail.com
3	FLORES	ROBERTO	flores2000@gmail.com
4	MANRIQUE	MARCELA	manrique.mar@gmail.com
5	PEREZ	ALEJANDRO	sur25perez@gmail.com

telefono			
<i>id</i>	<i>id_cliente</i>	<i>prefijo</i>	<i>numero</i>
1	3	0264	4245588
2	1	0261	4218877
3	2	0351	43355141
4	2	011	43377125
5	1	0264	4214277
6	5	0261	4285500
7	1	011	43281214

- ☐ La consulta generará un error de sintaxis porque le falta el RIGHT en la cláusula del JOIN.
- ☐ La consulta se ejecutará satisfactoriamente y devolverá 35 registros.
- ☐ La consulta generará un error de sintaxis porque le falta el LEFT en la cláusula del JOIN.



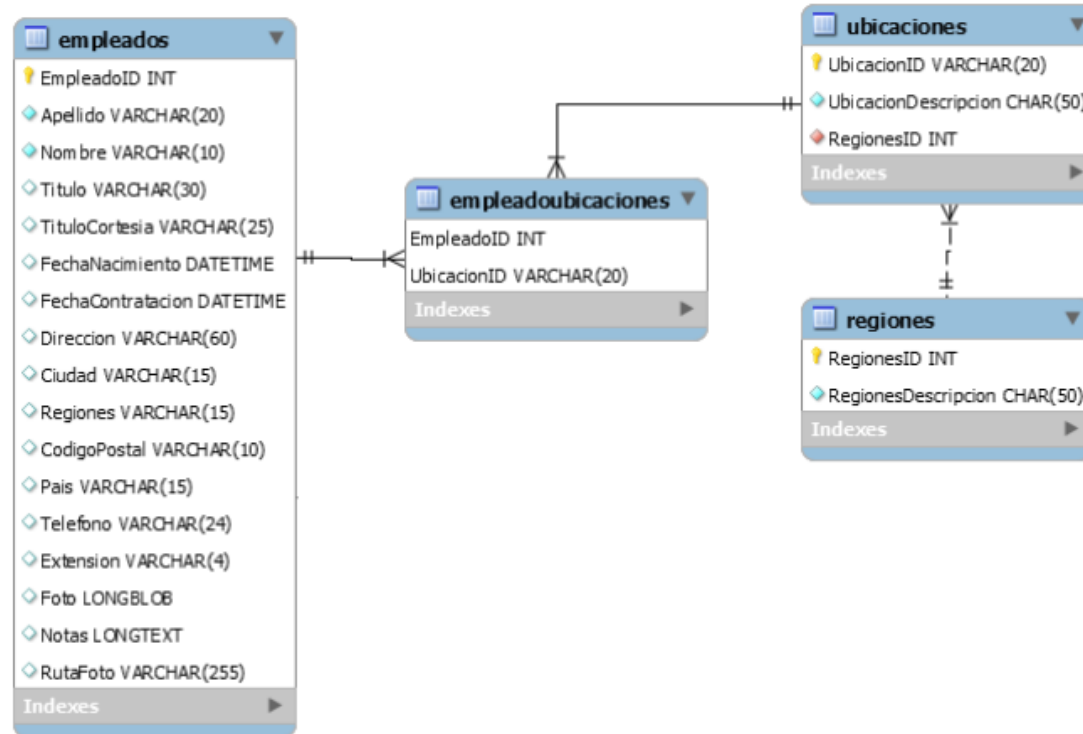
☒ Ninguna respuesta es correcta.

¿Cuál es la definición que está más relacionada a "Entidad"? \*

- ☒ Es un objeto, real o abstracto, acerca del cual se recoge información de interés para la base de datos.
- ☐ Describe solo las características principales de un Objeto, ejemplo: Nombre,Apellido, Edad
- ☐ Conjunto de uno o más atributos cuyos valores identifican unívocamente cada ocurrencia de la entidad.
- ☐ son la representación de la realidad no comprometida con ningún entorno informático: modelo entidad-relación.



Conseguir un listado de todos los empleados cuyo nombre empiece con la letra F \*  
y la descripción de la región sea "Southern"



- ☐ SELECT e.Apellido, e.Nombre, r.RegionesDescripcion FROM empleadoubicaciones eu INNER JOIN empleados e ON eu.EmpleadoID = e.EmpleadoID INNER JOIN ubicaciones u ON eu.UbicacionID = u.UbicacionID INNER JOIN regiones r ON u.RegionesID = r.RegionesID WHERE r.RegionesDescripcion LIKE "Southern" AND e.Nombre LIKE 'F\_';
- ☐ SELECT eu.Apellido, eu.Nombre, r.RegionesDescripcion FROM empleadoubicaciones eu INNER JOIN ubicaciones u ON eu.UbicacionID = u.UbicacionID INNER JOIN regiones r ON u.RegionesID = r.RegionesID WHERE r.RegionesDescripcion LIKE "Southern" AND e.Nombre LIKE '%F%';

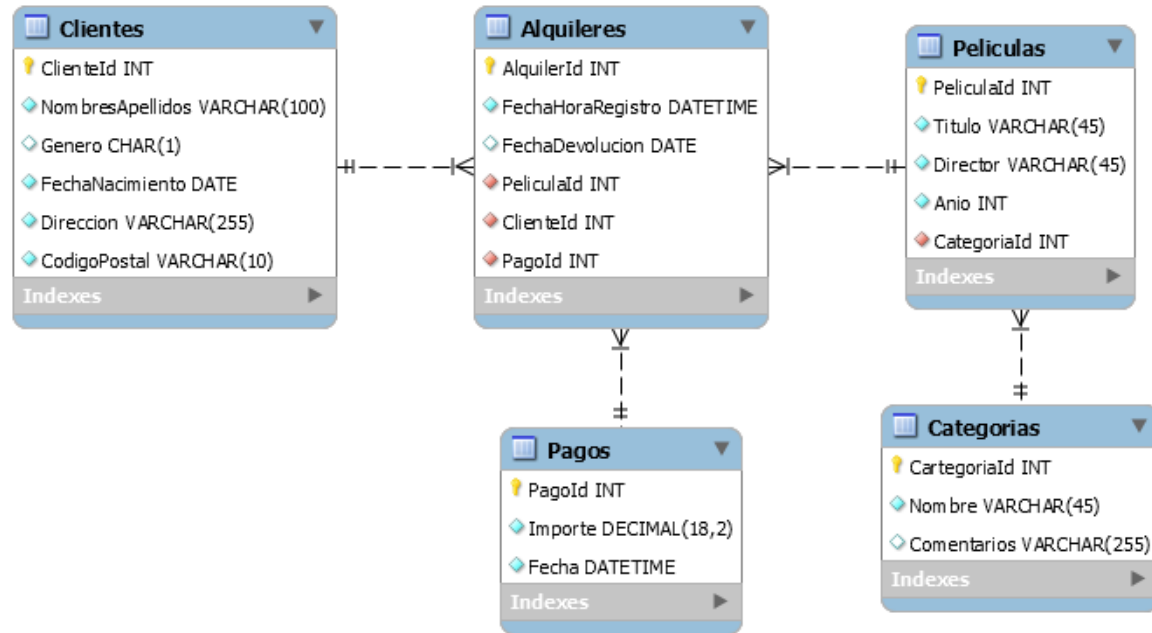


☐ SELECT e.Apellido, e.Nombre, r.RegionesDescripcion FROM empleadoubicaciones eu INNER JOIN empleados e ON eu.EmpleadoID = e.EmpleadoID INNER JOIN ubicaciones u ON eu.UbicacionID = u.UbicacionID INNER JOIN regiones r ON u.RegionesID = r.RegionesID WHERE r.RegionesDescripcion LIKE "Southern" AND e.Nombre LIKE '%F%';

☒ SELECT e.Apellido, e.Nombre, r.RegionesDescripcion FROM empleadoubicaciones eu INNER JOIN empleados e ON eu.EmpleadoID = e.EmpleadoID INNER JOIN ubicaciones u ON eu.UbicacionID = u.UbicacionID INNER JOIN regiones r ON u.RegionesID = r.RegionesID WHERE r.RegionesDescripcion LIKE "Southern" AND e.Nombre LIKE 'F%';



La siguiente instrucción inserta un registro en la tabla de pagos (tener en cuenta \* que la PK tiene la propiedad de AUTO\_INCREMENT): INSERT INTO pagos VALUES (DEFAULT, 100, '2021-05-18');

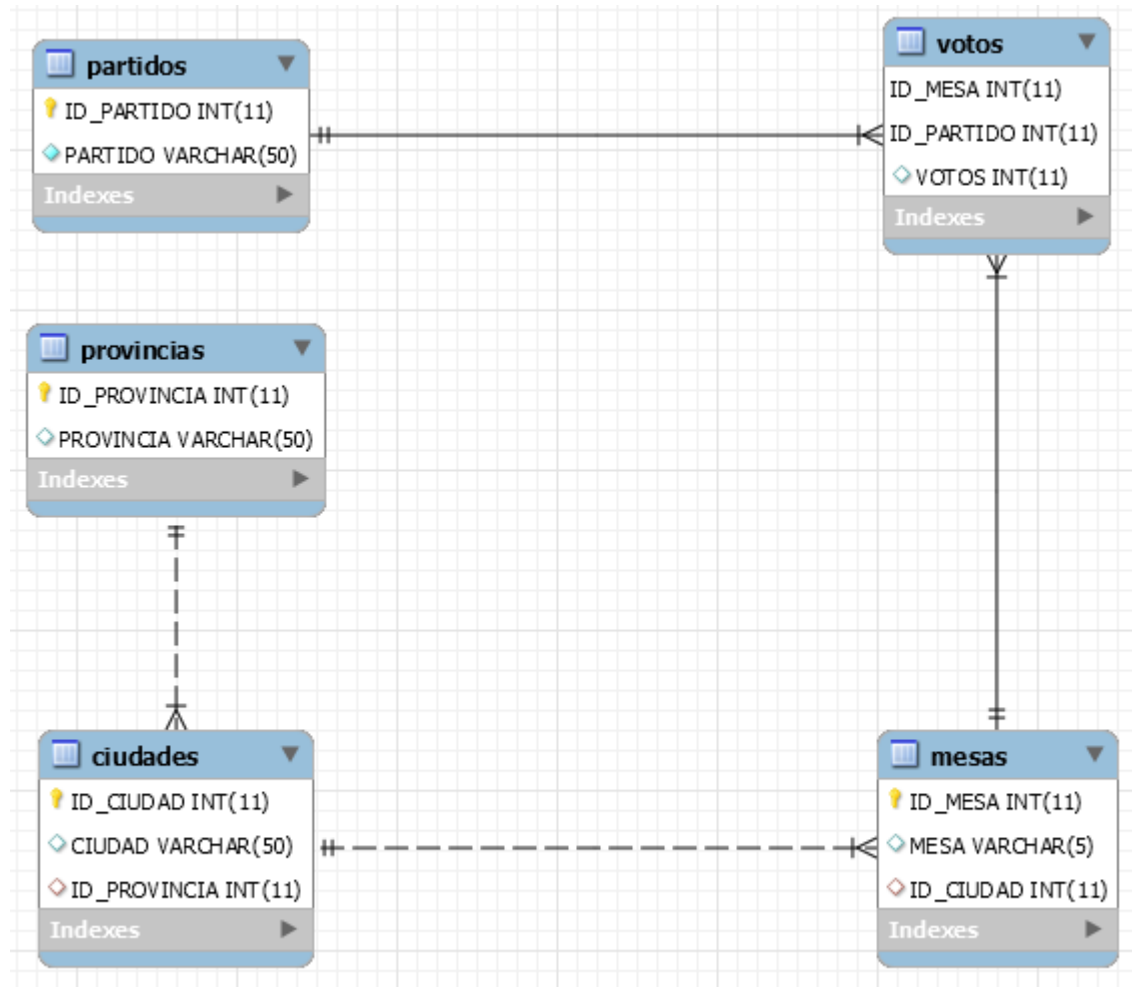


☒ VERDADERO

☐ FALSO



¿Cuál de las siguientes sentencias de SQL cuenta la cantidad promedio de votos \*  
para cada partido ordenado de menor a mayor por cantidad de votos?



- ☒ SELECT P.PARTIDO, AVG (V.VOTOS) as 'cantidad de votos del partido' FROM PARTIDOS as P INNER JOIN VOTOS as V ON P.ID\_PARTIDO = V.ID\_PARTIDO GROUP BY P.PARTIDO ORDER BY AVG(V.VOTOS) ASC





- ☐ SELECT P.PARTIDO, AVG (V.VOTOS) as 'cantidad de votos del partido' FROM PARTIDOS as P INNER JOIN VOTOS as V ON P.ID\_PARTIDO = V.ID\_PARTIDO GROUP BY V.VOTOS, P.PARTIDO ORDER BY AVG(V.VOTOS) DESC
- ☐ SELECT P.PARTIDO, AVG (V.VOTOS) as 'cantidad de votos del partido' FROM PARTIDOS as P INNER JOIN VOTOS as V ON P.ID\_PARTIDO = V.ID\_PARTIDO GROUP BY P.PARTIDO ORDER BY AVG(V.VOTOS) DESC
- ☐ SELECT P.PARTIDO, COUNT(V.VOTOS) as 'cantidad de votos del partido' FROM PARTIDOS as P INNER JOIN VOTOS as V ON P.ID\_PARTIDO = V.ID\_PARTIDO GROUP BY P.PARTIDO ORDER BY COUNT(V.VOTOS) ASC

La cláusula DISTINCT permite: \*

- ☒ Devolver valores únicos
- ☐ Traer valores distintos a uno solicitado pasado como parametro
- ☐ Unir datos en una sola columna
- ☐ Eliminar datos similares



Se necesita conocer el total del rating de las películas por cada uno de los géneros al que pertenecen ¿Cuál de las sentencias SQL es la correcta?

\*



- ☒ SELECT [g.name](#) 'Genero' ,sum(m.rating) "Total de Rating" FROM movies m INNER JOIN genres g ON m.genre\_id = [g.id](#) GROUP BY [g.name](#);
- ☐ SELECT [g.name](#) 'Genero' ,sum(m.rating) "Total de Rating" FROM movies m INNER JOIN genres g ON m.genre\_id = [g.id](#) ;
- ☐ SELECT [g.name](#) 'Genero' ,m.rating "Total de Rating" FROM movies m INNER JOIN genres g ON m.genre\_id = [g.id](#) GROUP BY [g.name](#);
- ☐ Ninguna de las anteriores



[Atrás](#)[Enviar](#)

Página 2 de 2

[Borrar formulario](#)

Nunca envíes contraseñas a través de Formularios de Google.

Este formulario se creó en Digital House. [Notificar uso inadecuado](#)

# Google Formularios



