

Examen de Base de Datos - 13

dc7juilland@gmail.com [Cambiar de cuenta](#)

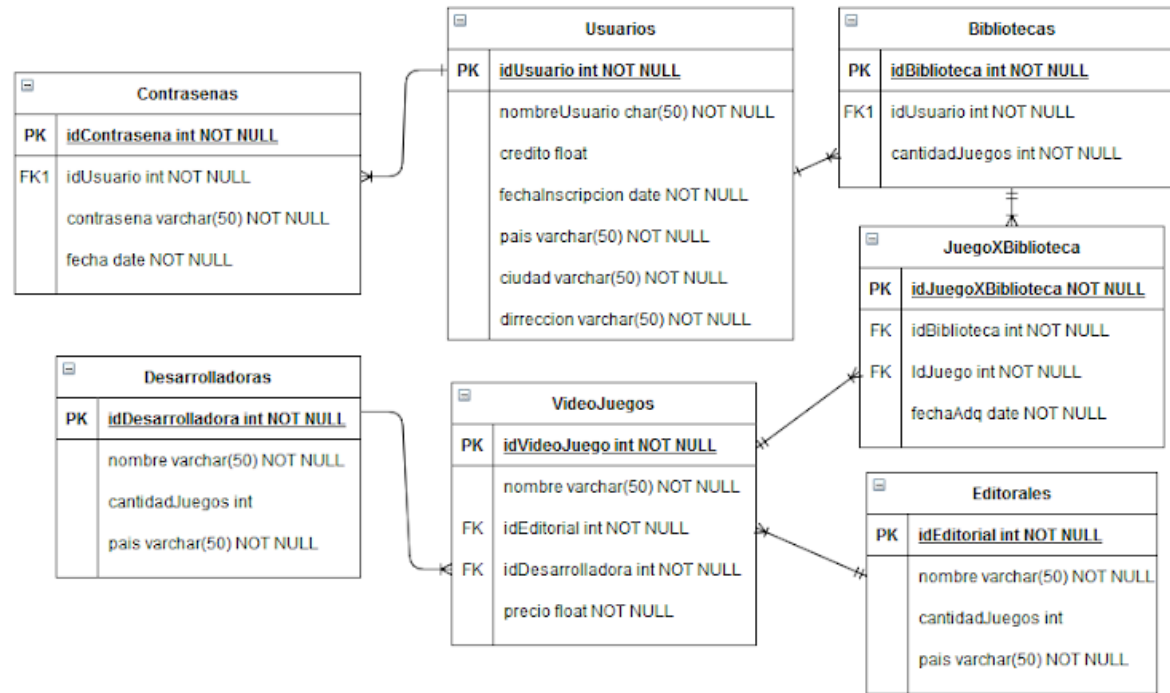


Borrador guardado

Preguntas



Basándonos en el siguiente DER. Que query nos permitirá traer el promedio de crédito de los usuarios según su país:



- ☐ SELECT pais, MAX(credito) as promedio FROM usuarios GROUP BY credito;
- ☐ SELECT pais, COUNT(credito) as promedio FROM usuarios GROUP BY pais;
- ☐ SELECT pais, AVG(credito) as promedio FROM usuarios GROUP BY credito;
- ☒ SELECT pais, AVG(credito) as promedio FROM usuarios GROUP BY pais;

Borrar selección



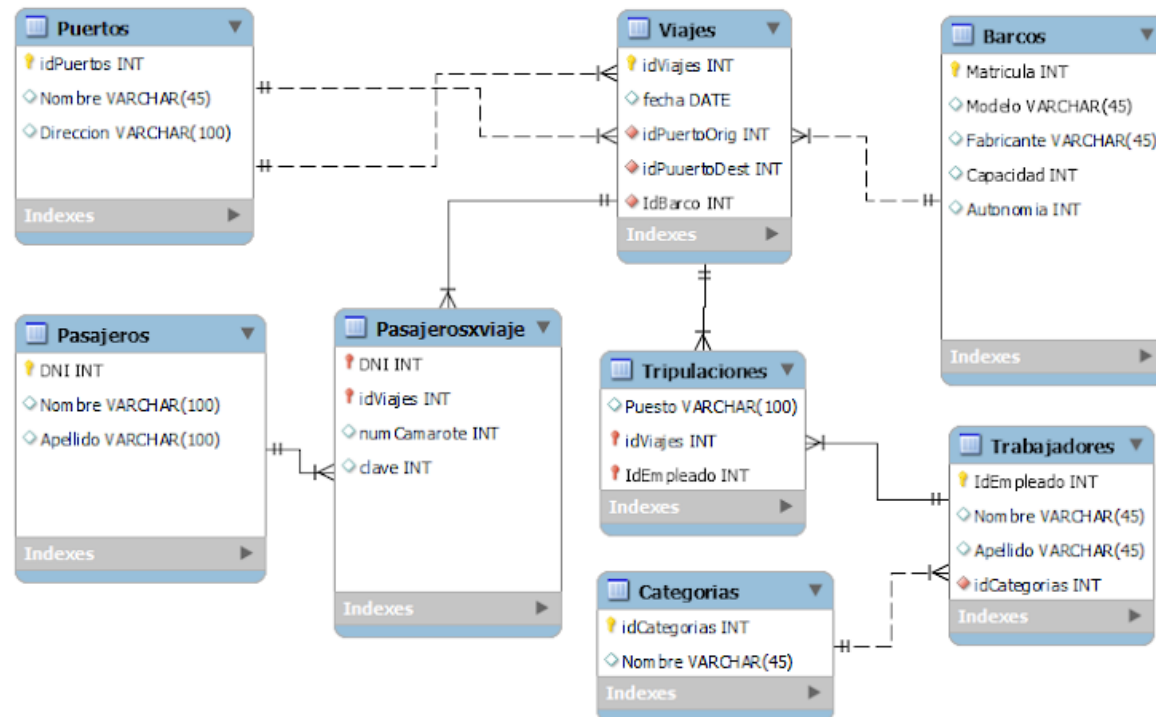
Se solicita crear una relación entre los empleados y sus categorías laborales. Cabe mencionar, que la fábrica asigna una sola categoría por empleado. ¿Qué tipo de relación es la apropiada?

- ☐ Uno a uno
- ☒ Muchos a uno
- ☐ Muchos a muchos

Borrar selección



Para listar los modelos de barcos y la menor capacidad por cada modelo, tenemos que ejecutar la siguiente query

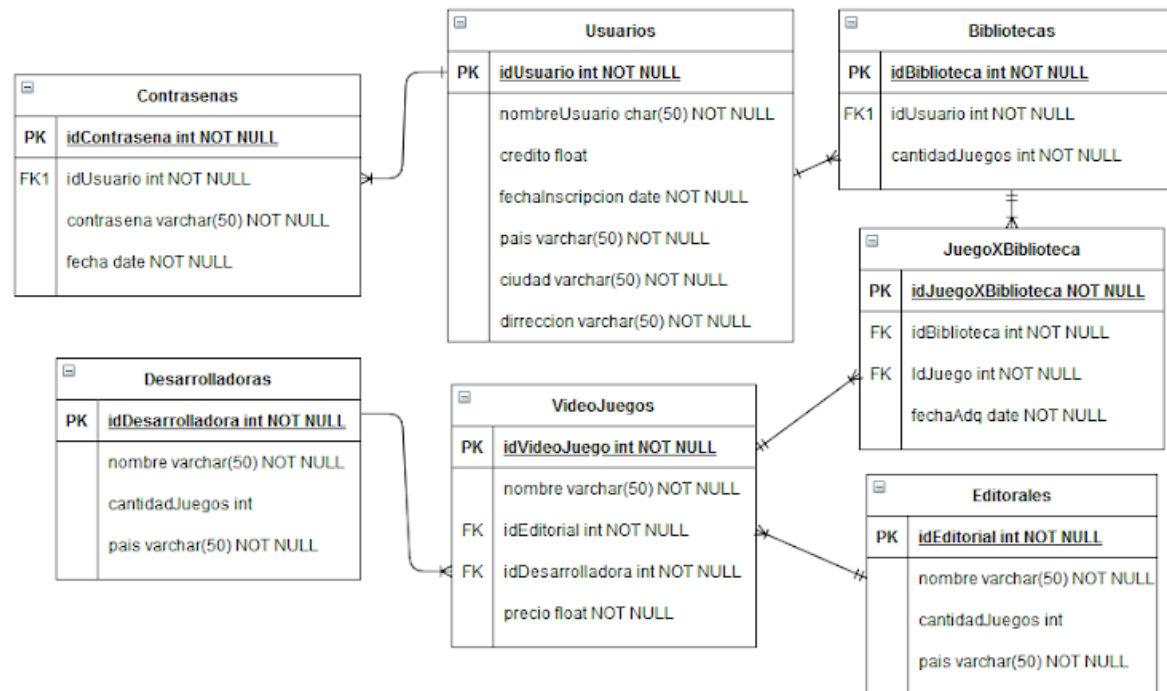


- ☒ SELECT modelo, MIN(capacidad) capacidad FROM barcos GROUP BY modelo ORDER BY MIN(capacidad) desc
- ☐ SELECT modelo, MAX(capacidad) capacidad FROM barcos GROUP BY modelo ORDER BY capacidad desc
- ☐ SELECT modelo, MAX(capacidad) capacidad FROM barcos ORDER BY capacidad desc
- ☐ Ninguna es correcta



Borrar selección

Basándonos en el siguiente DER. Esta query nos permite obtener un listado de los usuarios con más crédito de sus países: `SELECT nombreUsuario, pais, COUNT(credito) FROM usuarios GROUP BY pais;`

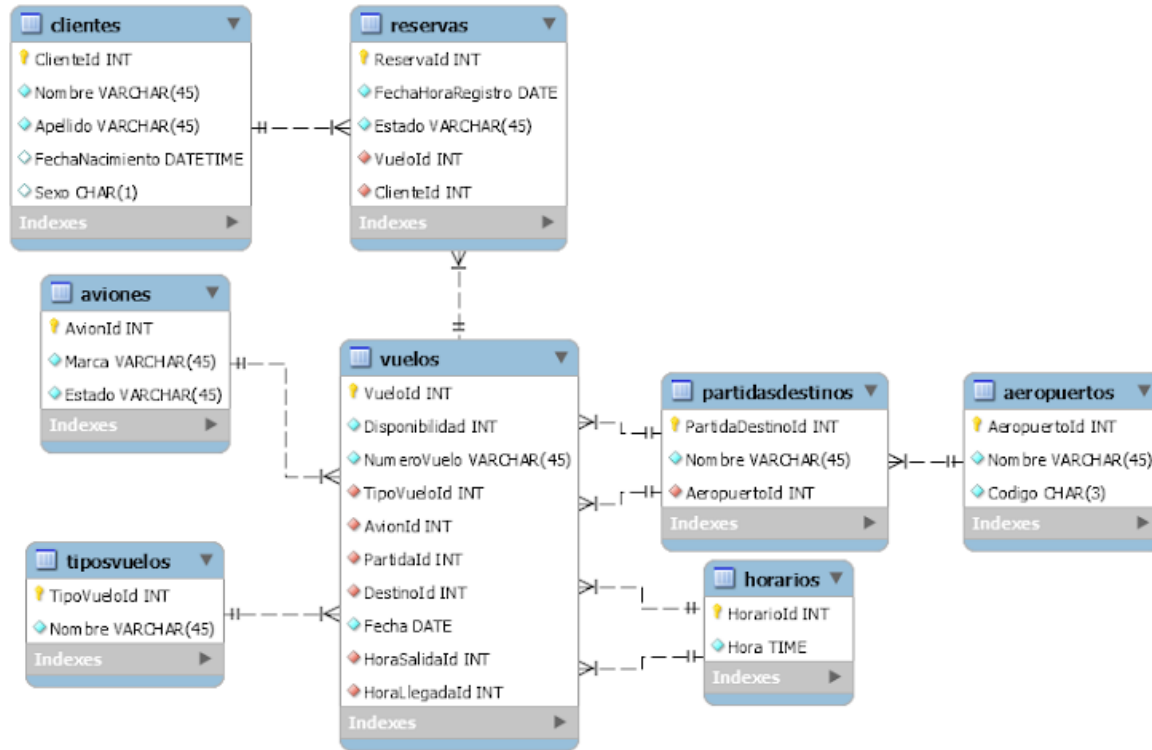

☐ VERDADERO

☒ FALSO

Borrar selección



¿Cuál de las siguientes sentencias de SQL lista en orden alfabético (sin repetir) las marcas de los aviones que tienen vuelo en la fecha 8/5/21?



- ☐ SELECT * FROM aviones INNER JOIN vuelos ON aviones.AvionId = vuelos.VueloId WHERE Fecha = '2021-04-08' 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.AvionId WHERE Fecha = '2021-05-08' ORDER BY Marca
- ☐ SELECT DISTINCT Marca FROM aviones INNER JOIN vuelos ON aviones.AvionId = vuelos.VueloId WHERE Fecha = '2021-04-08'



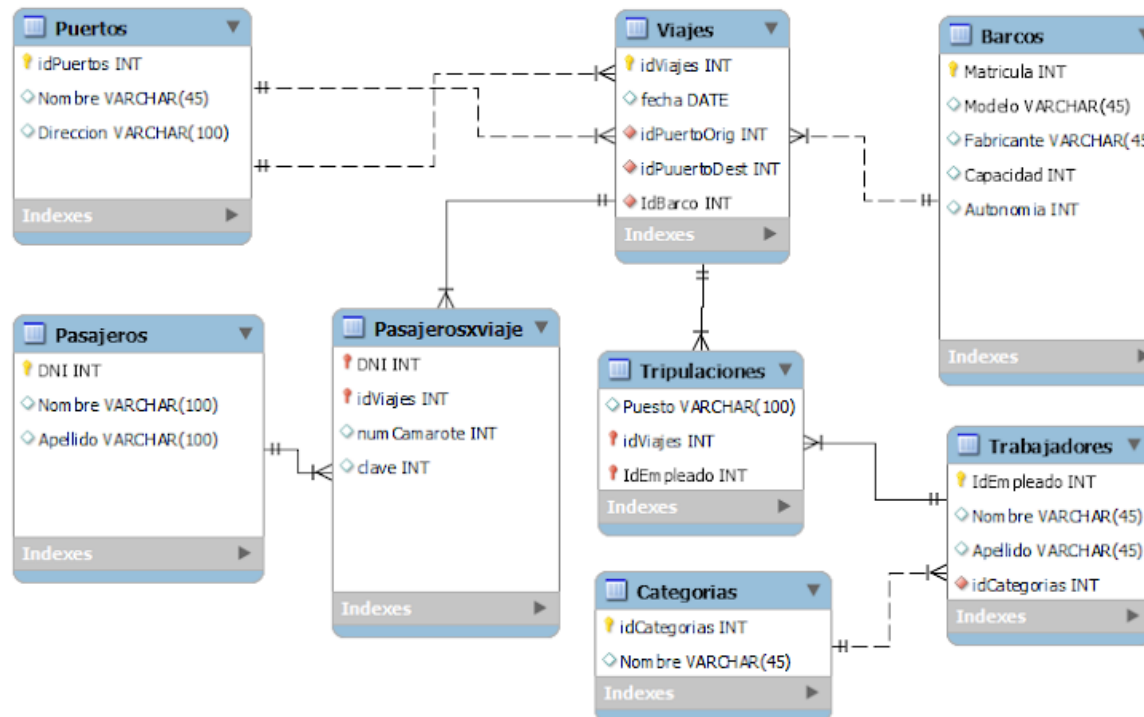
[Borrar selección](#)

La siguiente query `SELECT EXTRACT(YEAR FROM order_date) FECHA, count(*) as cantidad, sum(amount) as importe FROM orders GROUP BY fecha ORDER BY count(*) DESC LIMIT 1`

- ☒ Lista la cantidad de órdenes y el monto total del año que más órdenes tuvo.
- ☐ Lista la cantidad de órdenes y el monto total del año que más facturó.
- ☐ Lista la cantidad de órdenes y el monto total de la última orden ordenada por fecha.
- ☐ Lista la cantidad de órdenes y el monto total de la orden con mayor importe.

[Borrar selección](#)

Si queremos obtener un reporte de las fechas y el puerto origen del cual salen los barcos con capacidad mayor a 1000 tenemos que ejecutar la siguiente query:



- ☒ SELECT p.nombre AS nombre, v.fecha AS fecha FROM viajes v inner join puertos p on v.idPuertoOrig=p.idPuertos inner join barcos b ON v.idBarco = b.matricula where b.capacidad >1000
- ☐ SELECT p.nombre AS nombre, v.fecha AS fecha FROM viajes inner join barcos b ON v.idBarco = b.matricula where b.capacidad >1000
- ☐ SELECT p.nombre AS nombre, v.fecha AS fecha FROM puertos p on v.idPuerto=p.idPuerto inner join barcos b ON v.idBarco = b.matricula where b.capacidad >1000
- ☐ Ninguna es correcta



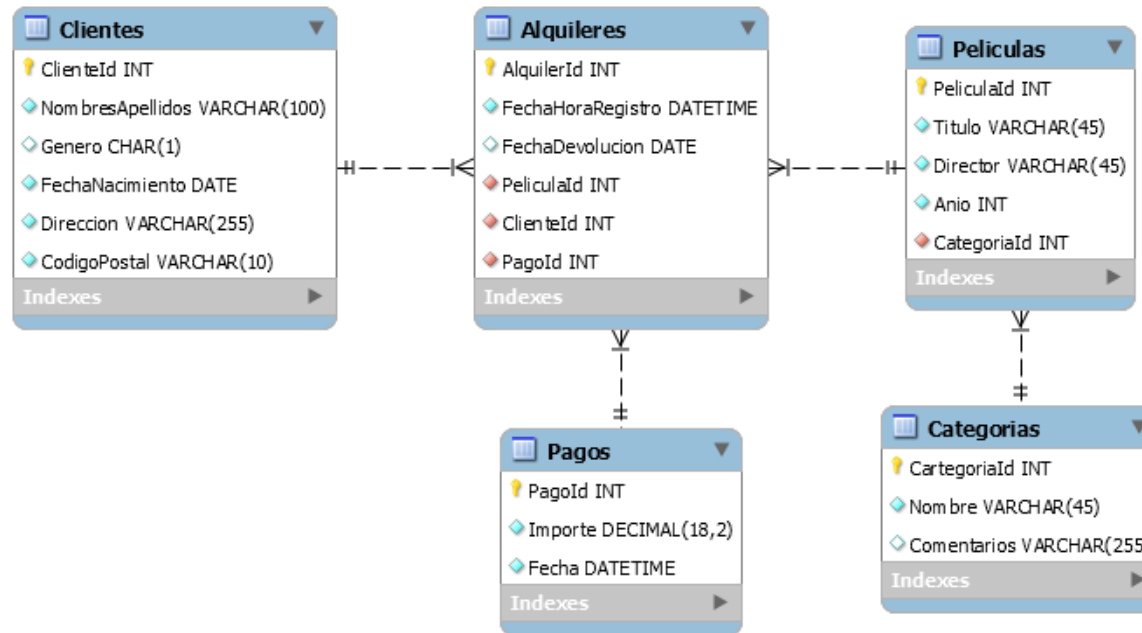
[Borrar selección](#)

Se solicita asignar un atributo para registrar el siguiente número 2,718281828459045. ¿Qué tipo de datos es el más apropiado?

- ☐ DOUBLE (1,15)
- ☒ DOUBLE (16,15)
- ☐ FLOAT (16,15)
- ☐ INT (16)

[Borrar selección](#)

¿Cuál de las siguientes sentencias de SQL lista todos los alquileres de películas de categoría del nombre Terror?



- ☐ SELECT * FROM alquileres INNER JOIN peliculas ON alquileres.PeliculaId = peliculas.PeliculaId WHERE Titulo LIKE 'Terror'
- ☐ SELECT * FROM alquileres INNER JOIN peliculasxalquiler ON alquileres.AlquilerId = peliculasxalquiler.AlquilerId INNER JOIN peliculas ON peliculasxalquiler.PeliculaId = peliculas.PeliculaId INNER JOIN categorias ON peliculas.CategoriaId = categorias.CategoriaId WHERE Nombre LIKE 'Terror'
- ☒ SELECT * FROM alquileres INNER JOIN peliculas ON alquileres.PeliculaId = peliculas.PeliculaId INNER JOIN categorias ON peliculas.CategoriaId = categorias.CategoriaId WHERE Nombre LIKE 'Terror'
- ☐ SELECT * FROM alquileres INNER JOIN peliculas ON alquileres.PeliculaId = peliculas.PeliculaId INNER JOIN categorias ON peliculas.CategoriaId = categorias.CategoriaId



[Borrar selección](#)

El resultado de un LEFT JOIN siempre tiene más filas que un INNER JOIN

- ☐ VERDADERO
- ☒ FALSO

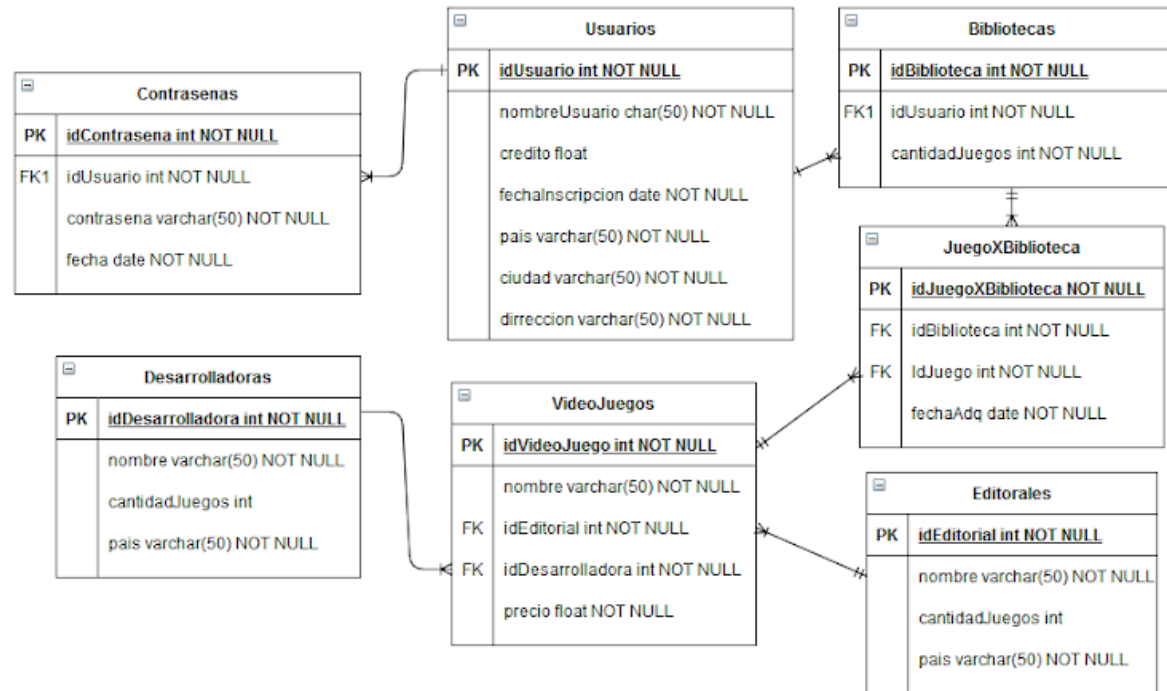
[Borrar selección](#)

Dentro de los tipos de datos FECHA existen...

- ☐ FULLTIME, DATETIME, TIMER, FULLDATE
- ☒ DATETIME, TIME, DATE
- ☐ FULLDATE, TIME, TIMEANDDATE
- ☐ DATE, DATEANDTYME, TINYDATE

[Borrar selección](#)

Basándonos en el siguiente DER. Si queremos obtener el nombre del usuario y crédito de aquel que tiene mayor crédito ¿Qué query utilizarías?

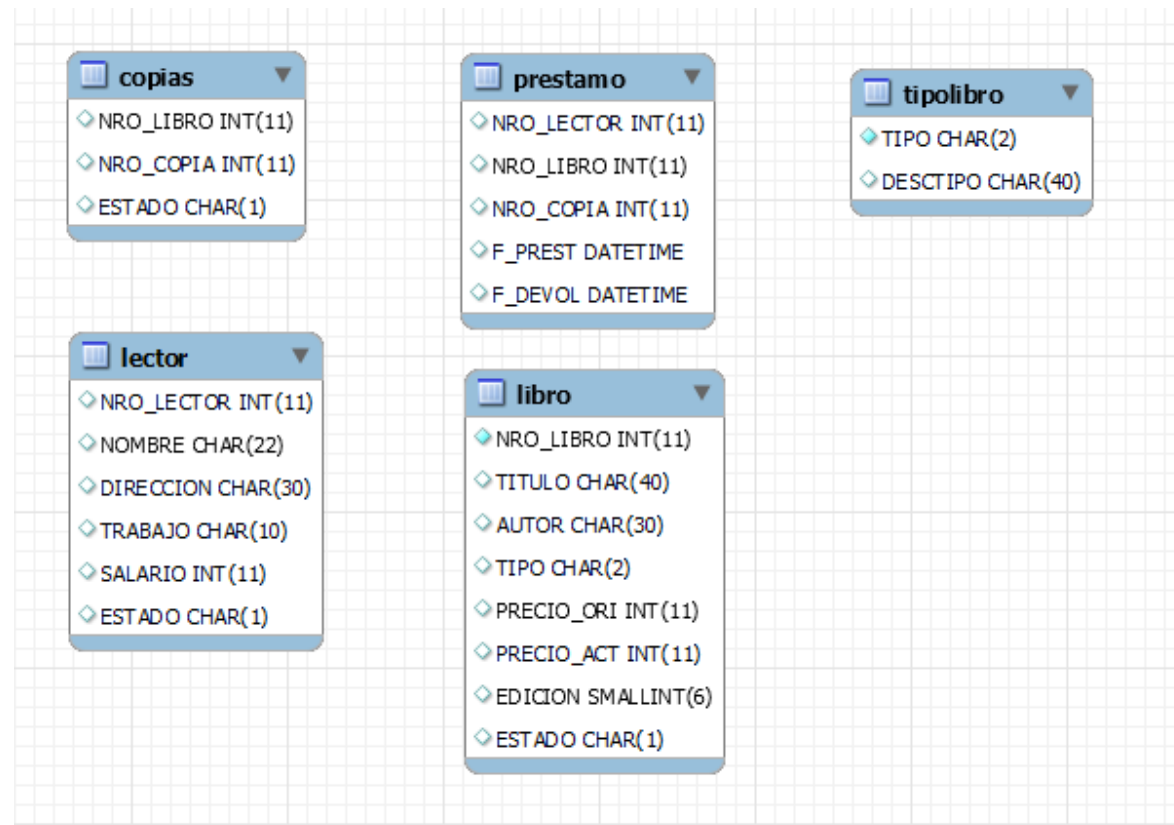


- ☒ SELECT nombreUsuario, credito FROM usuarios ORDER BY credito DESC LIMIT 1;
- ☐ SELECT nombreUsuario, credito FROM usuarios GROUP BY idUsuario HAVING MAX(credito) LIMIT 1;
- ☐ SELECT nombreUsuario, credito FROM usuarios GROUP BY idUsuario ORDER BY credito LIMIT 1;
- ☐ SELECT nombreUsuario, MAX(credito) FROM usuarios GROUP BY credito;

Borrar selección



Listar el Nro de Lector, Nombre y fecha de Préstamo de Todos los Lectores, hayan tenido Prestamos o no



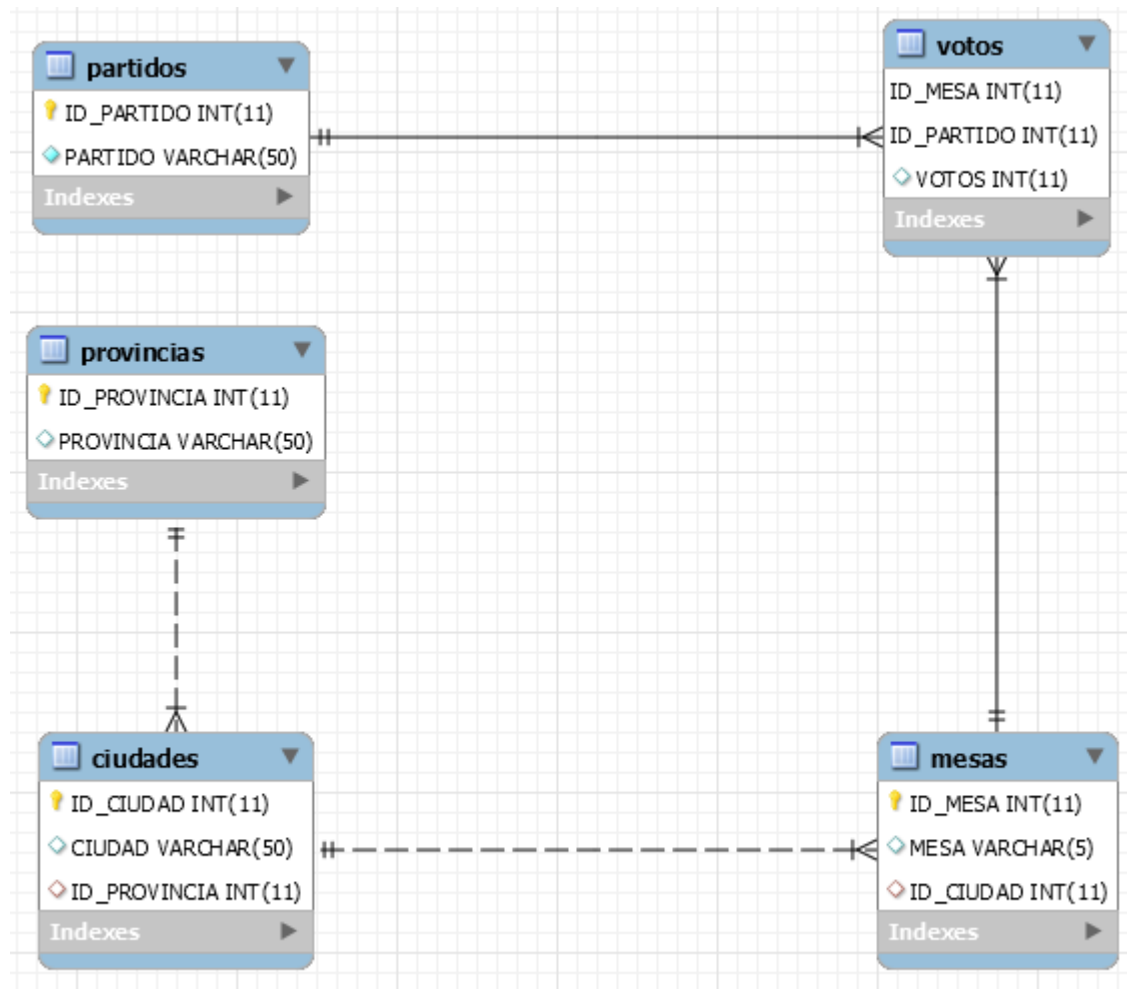
- ☒ SELECT lector.nro_lector,lector.NOMBRE,prestamo.f_prest FROM lector LEFT JOIN prestamo ON prestamo.nro_lector = lector.nro_lector
- ☐ SELECT lector.nro_lector,lector.NOMBRE,prestamo.f_prest FROM lector INNER JOIN prestamo ON prestamo.nro_lector = lector.nro_lector
- ☐ SELECT lector.nro_lector,lector.NOMBRE,prestamo.f_prest FROM lector RIGHT JOIN prestamo ON prestamo.nro_lector = lector.nro_lector



Borrar selección



¿Cuál de las siguientes sentencias de sql lista el id de la mesa con el nombre de su respectiva ciudad que pertenezcan a la provincia de "Buenos Aires"?



- ☐ SELECT M.ID_MESA, C.CIUDAD FROM MESAS as M INNER JOIN CIUDADES as C ON C.ID_CIUADAD = P.ID_CIUADAD INNER JOIN PROVINCIAS as P ON P.ID_PROVINCIA = P.ID_PROVINCIA WHERE P.PROVINCIA LIKE 'Buenos Aires'



- ☐ SELECT M.ID_MESA, C.CIUDAD FROM MESAS as M JOIN CIUDADES as C ON C.ID_CIUDAD = M.ID_CIUDAD JOIN PROVINCIAS as P ON P.ID_PROVINCIA = P.ID_PROVINCIA WHERE P.PROVINCIA LIKE 'Buenos Aires'
- ☐ SELECT M.ID_MESA, C.CIUDAD FROM MESAS as M INNER JOIN CIUDADES as C ON C.ID_CIUDAD = M.ID_CIUDAD INNER JOIN PROVINCIAS as P ON P.ID_PROVINCIA = P.ID_PROVINCIA WHERE P.PROVINCIA LIKE 'Buenos Aires'
- ☒ Ninguna es correcta

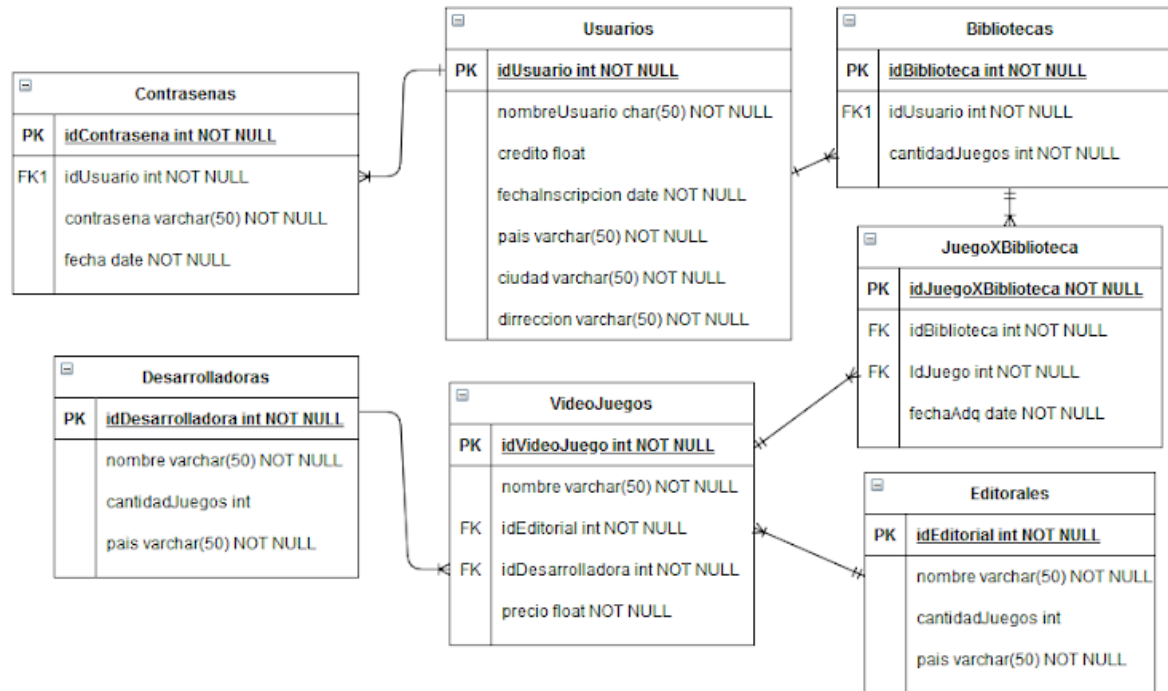
[Borrar selección](#)

Se solicita crear una relación entre los empleados de una fábrica y sus turnos laborales. La fábrica tiene 3 turnos rotativos. ¿Qué tipo de relación es la apropiada?

- ☒ Muchos a muchos
- ☐ Uno a uno
- ☐ Uno a muchos

[Borrar selección](#)

Basándonos en el siguiente DER. Si queremos obtener un listado de los usuarios cuyo país sea 'Argentina' y su ciudad 'Buenos Aires' la query sería:



- ☐ SELECT * FROM usuarios WHERE pais = "Argentina", ciudad = "Buenos Aires";
- ☐ SELECT * FROM usuarios WHERE pais(Argentina) AND ciudad("Buenos Aires");
- ☒ SELECT * FROM usuarios WHERE pais = "Argentina" AND ciudad = "Buenos Aires";
- ☐ SELECT * FROM usuarios WHERE pais(Argentina), ciudad("Buenos Aires");

Borrar selección



El LEFT JOIN, al igual que el RIGHT JOIN, devuelve todos los registros de una de las tablas relacionadas cumplan o no la condición indicada.

- ☒ Verdadero
- ☐ Falso
- ☐ Solo si las tablas tienen la misma cantidad de campos

Borrar selección

Si queremos insertar, actualizar o eliminar registros ¿Qué funciones utilizaríamos?

- ☒ insert, update y delete
- ☐ create, drop y alter
- ☐ select, insert, delete y update
- ☐ Ninguna de las anteriores

Borrar selección



Queremos obtener los empleados que ingresaron a la empresa entre los años 2010 y 2012 para un informe. ¿Qué consulta consideras que es la correcta?

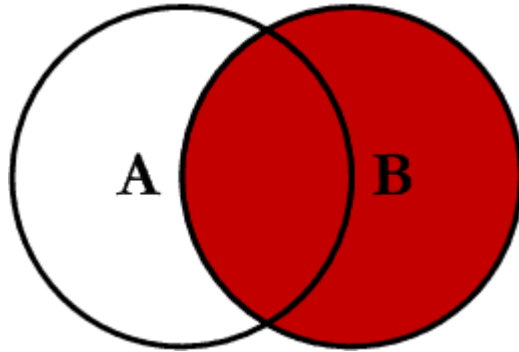
empleado	
PK	<u>id</u>
	Nombre
	Apellido
	DNI
	Legajo
	Puesto
	Fecha_Ingreso

- ☐ SELECT * FROM empleado WHERE fecha_ingreso LIKE '2010' OR '2012'
- ☐ SELECT * FROM empleado WHERE fecha_ingreso BETWEEN '2010' AND '2012'
- ☒ SELECT * FROM empleado WHERE fecha_ingreso BETWEEN '2010-01-01' AND '2012-12-31'
- ☐ SELECT * FROM empleado WHERE fecha_ingreso BETWEEN '2010/01/01' OR '2012/12/31'

Borrar selección



La siguiente imagen representa a



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

Borrar selección

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



