

NORMALIZACIÓN [1,5 PUNTOS]

Normaliza la siguiente relación hasta 3ª FN, para ello, rellena la tabla que se adjunta:

PEDIDO (id_pedido, id_cliente, id_comercial, teléfono, cantidad, fecha, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, categoría, comisión, localidad, provincia)

Donde:

- cantidad es la cantidad de productos que compra el cliente
- fecha es la fecha en la que se realiza la compra
- teléfono es el teléfono del cliente
- categoría es la categoría del cliente
- comisión es la comisión que le corresponde al comercial

PRIMERA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN
La relación ya cumple esta forma, ya que en cada campo de cada tupla solo hay un dato.

¿Por qué no se encuentra en 1ª FN?

Indica las relaciones resultantes del proceso de normalizar a 1ª FN:
PEDIDO (id_pedido, id_cliente, id_comercial, teléfono, cantidad, fecha, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, categoría, comisión, localidad, provincia)

SEGUNDA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN
PEDIDO (id_pedido, id_cliente, id_comercial, teléfono, cantidad, fecha, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, categoría, comisión, localidad, provincia)

¿Por qué no se encuentra en 2ª FN? Indica las dependencias, si las hay
Porque hay atributos que no dependen de la clave primaria en su totalidad.

telefono, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, categoria, localidad, provincia → id_cliente

nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, comisión
→ id_comercial

Indica las relaciones resultantes del proceso de normalizar a 2ª FN:

PEDIDO (id_pedido, id_cliente, id_comercial, cantidad, fecha)

CLIENTE(id_cliente, telefono, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, categoria, localidad, provincia)

COMERCIAL(id_comercial, nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, comisión)

TERCERA FORMA NORMAL

Indica la relación o relaciones que no cumplen esta FN

CLIENTE(id_cliente, telefono, nombre_cliente, apellido1_cliente, apellido2_cliente, ciudad_cliente, categoria, localidad, provincia)

¿Por qué no se encuentra en 3ª FN? Indica las dependencias, si las hay
No se encuentra en tercera forma ya que hay atributos que no son clave que dependen de otros atributos que son clave.
provincia, ciudad_cliente → Localidad

Indica las relaciones resultantes del proceso de normalizar a 3ª FN:

CLIENTE(id_cliente, telefono, nombre_cliente, apellido1_cliente, apellido2_cliente, categoria, localidad)

DIRECCION(localidad, provincia, ciudad_cliente)

ESQUEMA RELACIONAL NORMALIZADO

Indica las relaciones resultantes después de haber normalizado hasta 3ª FN:

PEDIDO (id_pedido, id_cliente, id_comercial, cantidad, fecha)

CLIENTE(id_cliente, telefono, nombre_cliente, apellido1_cliente, apellido2_cliente, categoria, localidad)

DIRECCION(localidad, provincia, ciudad_cliente)

COMERCIAL(id_comercial, nombre_comercial, apellido1_comercial, apellido2_comercial, ciudad_comercial, comisión)

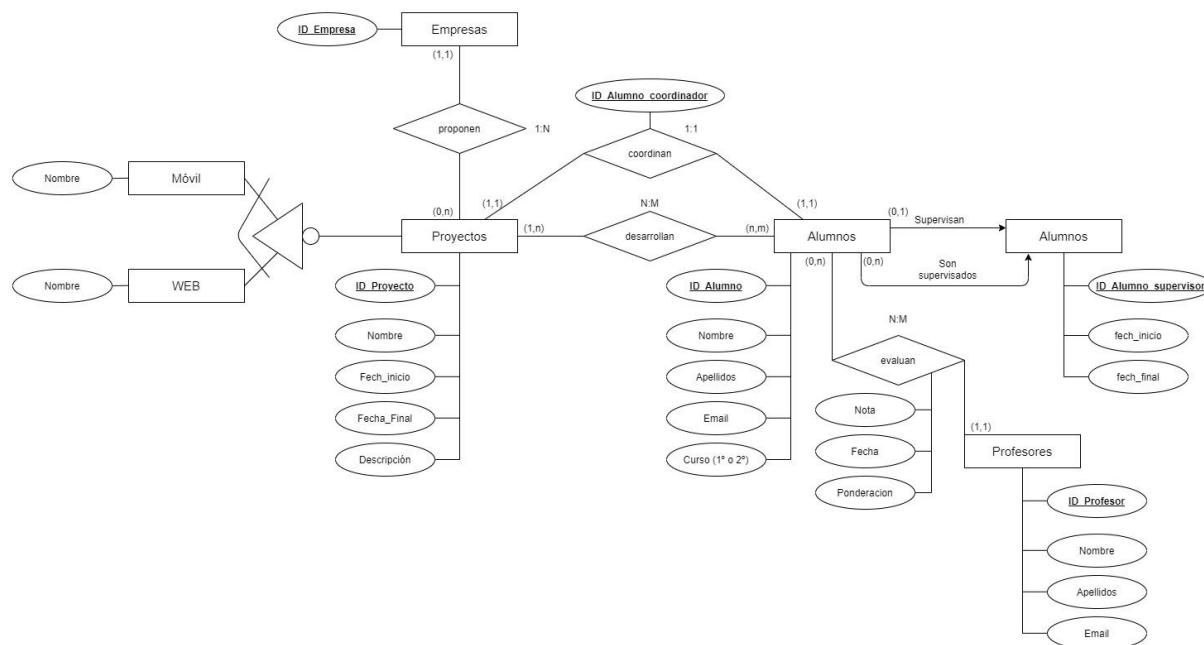
MODELO ENTIDAD RELACIÓN [4,5 PUNTOS]

El departamento de Informática del IES Alixar desea diseñar una base de datos para gestionar los alumnos que participan en los proyectos de investigación que se han propuesto. Se trata de proyectos que se van a realizar entre alumnos de primer y segundo curso del ciclo de DAW.

- De cada proyecto se desea almacenar un identificador único, nombre, fecha de inicio, fecha de finalización y una descripción.
- En el instituto colaboramos con distintas empresas, las cuales se encargan de proponer esos proyectos. Cada empresa puede proponer hasta varios de nuestros proyectos.
- Ten en cuenta que estos proyectos pueden ser de aplicaciones web o móvil. En caso de ser una aplicación web, queremos guardar el dominio en el que se va a alojar y en caso de ser móvil, guardaremos para qué entorno se ha desarrollado (que puede ser Android o IOS).
- Los alumnos se organizan siempre en grupo para trabajar en los proyectos durante un periodo de tiempo, determinado por una fecha de inicio y una fecha de fin. Tenga en cuenta que un mismo alumno puede trabajar en el mismo proyecto en diferentes épocas.
- De cada alumno se desea almacenar un identificador único, nombre, apellidos y email.
- Los alumnos que participan en los proyectos pueden ser de primer o segundo curso.
- Un alumno puede ser supervisado por otro. Deberemos almacenar los alumnos que han supervisado y durante qué periodo lo han hecho. Ten en cuenta que el tiempo de supervisión viene determinado por una fecha de inicio y una fecha de fin.
- De todos los alumnos que trabajan en el proyecto hay uno que es el desarrollador principal, que será el encargado de coordinar el proyecto. Es necesario almacenar quién es el desarrollador principal de cada uno de los proyectos. Tenga en cuenta que siempre habrá un único desarrollador principal por cada proyecto y que un alumno solo podrá coordinar un proyecto.
- Los alumnos son evaluados por profesores. De cada profesor guardaremos un identificador único, nombre, apellidos y email.
- De esta evaluación se guardará la fecha, la nota y la ponderación correspondiente.
- Otros aspectos a tener en cuenta:
 - Habrá alumnos que no tengan a otro alumno que los supervise y no todos supervisan a alguien.
 - Si algún alumno se ofrece y el profesorado lo estima oportuno, podría trabajar en varios proyectos a la vez.
 - En la base de datos guardaremos todas las empresas de la zona, puede que alguna de ellas no financie ninguno de nuestros proyectos.

Debes hacer:

1. Diseño del Modelo Entidad Relación que cumpla todos estos requisitos
2. Explicar:
 - a. Las cardinalidades de todas las relaciones que has considerado
 - a. El motivo de usar jerarquías o entidades débiles, si es el caso.



Cardinalidades:

Empresas - Proponen - Proyectos → Una empresa puede proponer ningún o varios proyectos, debido a que aunque no financie un proyecto queremos guardar esa empresa en la base de datos. Y un mismo proyecto solo puede ser propuesto por una sola empresa.

Proyectos - Desarrollan - Alumnos → Un alumno siempre va a participar al menos en un proyecto, y si los profesores lo vieran oportuno podría participar en varios, por ello la cardinalidad es (1,n). Y en un proyecto siempre van a participar varios alumnos por ello la cardinalidad es (n,m)

Alumnos - Evalúan - Profesores → Un alumno siempre va a ser evaluado, por ello es necesario que un profesor lo evalúe y sólo podrá ser evaluado por un profesor, por ello la cardinalidad es (1,1). Y un profesor no tiene porqué evaluar, puede ser simplemente colaborador, o puede evaluar a varios alumnos, por ello la cardinalidad es (0,n)

Alumnos - Coordinan - Proyectos → Un proyecto solo puede ser coordinado por un solo alumno, pero a su vez un alumno solo puede coordinar un proyecto al mismo tiempo, por ello ambas cardinalidades son (1,1).

Reflexiva Alumnos - Alumnos → Habrá alumnos que no tengan a otro alumno que los supervise y no todos supervisan a alguien, por ello un alumno puede supervisar a varios o ningún alumno (0,n) y un alumno puede o no ser supervisado (0,1).

La jerarquía es necesaria para diferenciar los dos tipos de proyectos ya que cada uno tiene sus atributos. En este caso la jerarquía es total ya que todos los proyectos van a pertenecer a uno de los dos tipos, y además es exclusiva ya que un proyecto o es Web o es Móvil.

Para el curso de los alumnos no he usado jerarquía, he usado un atributo que tendrá valor 1 si el alumno es de primer curso o 2 si es de segundo curso, ya que suponemos que un alumno terminará el proyecto antes de cambiar de curso.

He guardado la fecha, ponderación y la nota de cada alumno dentro de evalua, ya que veo necesario que sepamos a qué alumno se evalúa y qué profesor es el que evalúa, creando así una tabla en el modelo relacional que reflejaría el alumno y el profesor además de la fecha, ponderación y la nota de esa evaluación.

La relación coordina contiene el ID del alumno coordinador.

CONSULTAS [4 PUNTOS]

Para realizar estas consultas, debes:

- Usar la BD que tienes a tu disposición (jardineria.sql)
- Mostrar únicamente la información que se pide y en el orden pedido
- Seguir las reglas sintácticas y de formato que hemos visto en clase:
 - Uso de mayúsculas y minúsculas
 - Saltos de línea
- Copiar en cada apartado la consulta correspondiente

A partir de la base de datos "jardineria", contesta:

1. Devuelve un listado con el código de pedido, código de cliente, fecha esperada y fecha de entrega de los pedidos que no han sido entregados a tiempo. SELECT codigo_pedido, codigo_cliente, fecha_esperada, fecha_entrega FROM pedido WHERE fecha_entrega>fecha_esperada;	0,3
2. Devuelve un listado de todos los pedidos que fueron rechazados en 2009. SELECT * FROM pedido WHERE estado='rechazado' AND YEAR(fecha_entrega)=2009;	0,3
3. Devuelve un listado con el nombre, la gama, el proveedor, el precio de venta y la cantidad que hay en stock de los productos que pertenecen a la gama Ornamentales y que tienen más de 100 unidades en stock. El listado deberá estar ordenado por su precio de venta, mostrando en primer lugar los de mayor precio. SELECT nombre, gama, proveedor, precio_venta, cantidad_en_stock FROM producto WHERE gama='Ornamentales' AND cantidad_en_stock>100 ORDER BY precio_venta DESC;	0,3
4. Devuelve un listado con el código del cliente, su nombre y su ciudad de los clientes que sean de la ciudad de Madrid y cuyo representante de ventas tenga el código de empleado 11 o 30. SELECT codigo_cliente, nombre_cliente, ciudad, codigo_empleado_rep_ventas FROM cliente WHERE ciudad='Madrid' AND (codigo_empleado_rep_ventas=11 OR codigo_empleado_rep_ventas=30);	0,3

<p>5. Muestra el nombre de los clientes que viven en la misma ciudad que donde se encuentra la oficina 'MAD-ES'</p> <pre> SELECT nombre_cliente FROM cliente WHERE ciudad=(SELECT ciudad FROM oficina WHERE codigo_oficina='MAD-ES') ; </pre>	0,3
<p>6. Devuelve un listado con el nombre de la gama de todos los productos existentes en la base de datos y el número de productos que tiene cada uno. El resultado deberá estar ordenado alfabéticamente por el nombre de la gama (de la A a la Z). Ponle un alias adecuado a la columna correspondiente al número de productos.</p> <pre> SELECT gama, COUNT(codigo_producto) AS NumeroProductos FROM producto GROUP BY gama ORDER BY gama; </pre>	0,4
<p>7. Devuelve el código del producto y la cantidad que hay en stock del producto que más unidades tiene en stock.</p> <pre> SELECT codigo_producto, cantidad_en_stock FROM producto WHERE cantidad_en_stock=(SELECT MAX(cantidad_en_stock) FROM producto) ; </pre>	0,3
<p>8. Calcula cuántos empleados hay en cada puesto. El resultado sólo debe mostrar dos columnas, una con el nombre del puesto y otra con el número de empleados que hay en ese puesto (esta columna la debes llamar 'num_empleados'). El resultado deberá estar ordenado de mayor a menor número de empleados.</p> <pre> SELECT puesto, COUNT(codigo_empleado) AS num_empleados FROM empleado GROUP BY puesto ORDER BY num_empleados DESC; </pre>	0,4
<p>9. Devuelve el nombre, apellido1 y puesto de los empleados que no representen a ningún cliente.</p> <pre> SELECT nombre, apellido1, puesto FROM empleado WHERE codigo_empleado NOT IN (SELECT codigo_empleado_rep_ventas FROM cliente) ; </pre>	0,6

<p>10. Calcula la media del importe de los pagos realizados en cada año. El resultado sólo debe mostrar dos columnas, una con el año y otra con la media de éste. Ponle un alias adecuado a cada columna y muestra los datos ordenados de mayor a menos importe.</p> <pre>SELECT YEAR(fecha_pago) AS Año, AVG(total) AS Media_Pagos FROM pago GROUP BY Año ORDER BY Media_Pagos DESC;</pre>	0,4
<p>11. Devuelve el nombre, los apellidos y el email de los empleados que están a cargo de Alberto Soria.</p> <pre>SELECT nombre, apellido1, apellido2, email FROM empleado WHERE codigo_jefe=(SELECT codigo_empleado FROM empleado WHERE nombre='Alberto' AND apellido1='Soria') ;</pre>	0,4