

APELLIDOS: SOLUCIONES

OPCIÓN: A

NOMBRE: _____

FECHA: 25/01/23

Curso: 2022/23

Convocatoria: 1 Evaluación

Módulo: Base de Datos

Ciclo Formativo: Seleccionar el que proceda

- Desarrollo de aplicaciones multiplataforma
- Desarrollo de aplicaciones web

Instrucciones.

- La prueba consta de 10 puntos.
- Cada ejercicio o sub-ejercicio tiene su puntuación.
- Debe realizar con bolígrafo azul o negro, el incumplimiento de esta norma implicará la calificación de 0 puntos en el examen.
- El DNI tiene que estar sobre la mesa.
- Se dispone de 2 horas para la realización.
- Se debe responder en el lugar habilitado para ello.
- En caso de necesitar más folios, se debe indicar el número de hojas que se entrega adicionales al enunciado del examen. Estas deberán numerarse y escribir claramente el nombre del alumno y el ejercicio que se está resolviendo.

Entrego ____ hojas adicionales al examen.

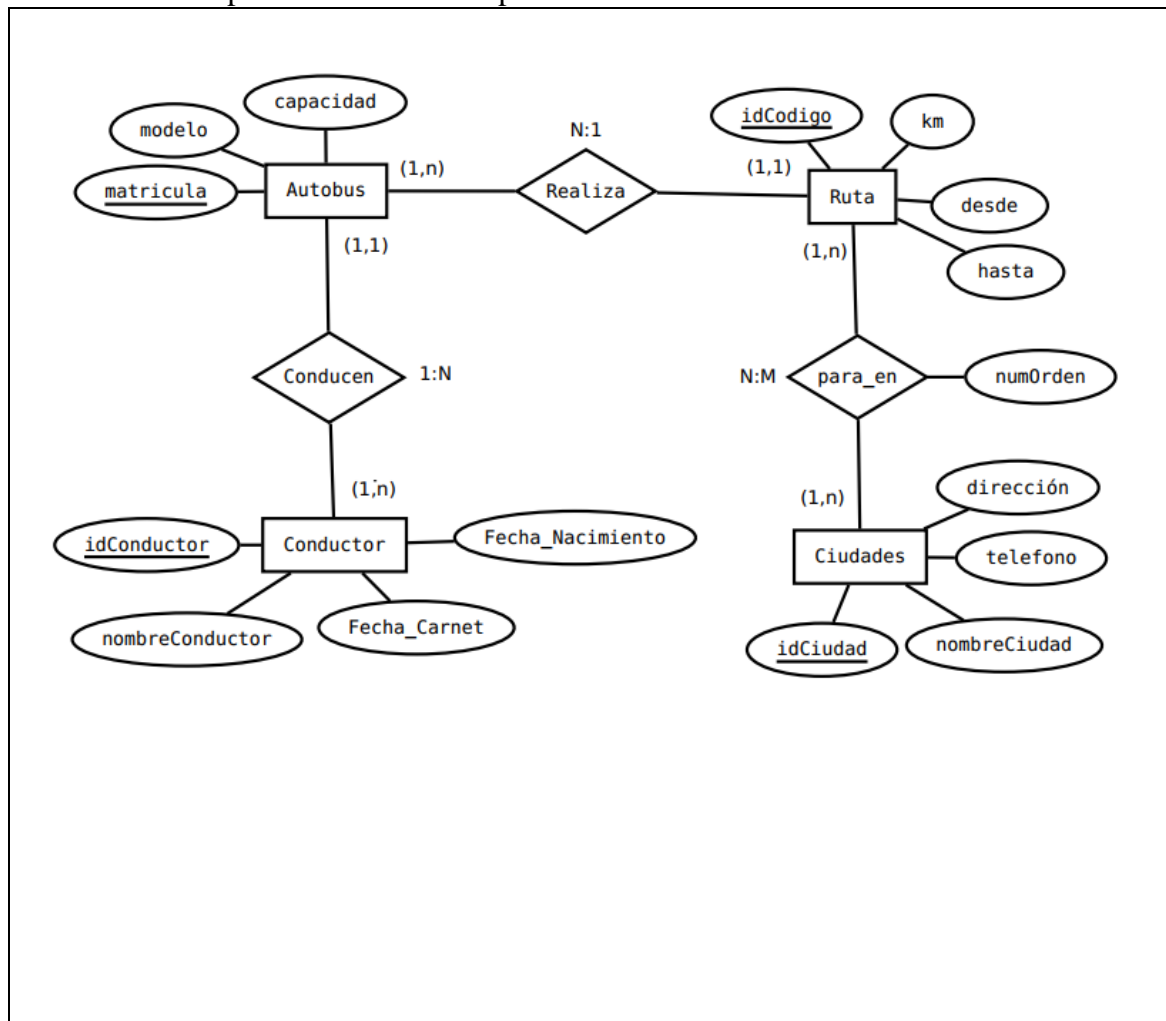
Verificado por el profesor:

● **SUERTE**

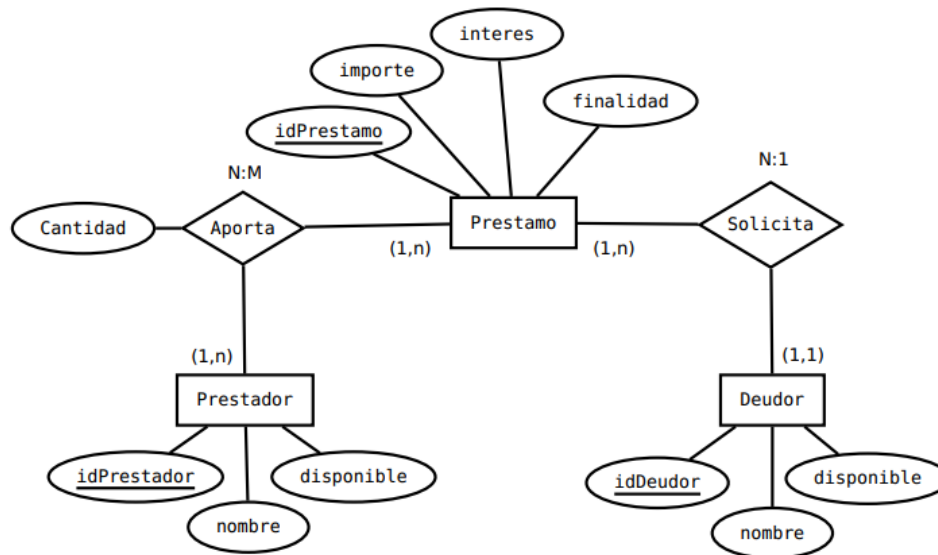
Ejercicio 1. Construye el esquema entidad relación. (2 puntos)

Una empresa de transporte desea crear una base de datos para almacenar información sobre sus rutas.

- La empresa dispone de una flota de autobuses que distribuyen en una serie de rutas. En cada ruta, el autobús pasa por un conjunto de ciudades en las que tiene paradas.
- Una ruta se identifica por un código y se caracteriza por los km. Totales de recorrido, el origen y el destino final. De cada autobús, se almacena su matrícula, el modelo, su capacidad (plazas) y el nombre del conductor. Se asume que un autobús solo puede estar realizando una ruta.
- Las rutas tienen paradas en distintas ciudades. De cada ciudad, almacenaremos el nombre, junto con la dirección y el teléfono del lugar de parada. Para organizar las rutas, cada parada tiene un número de orden, que puede variar entre distintas rutas (una misma ciudad puede pertenecer a varias rutas).
- Del conductor, nos interesa almacenar el nombre, fecha de carnet y la fecha de nacimiento, los conductores estarán identificados mediante un código interno.
- Por último, se asume que un conductor solo puede conducir un autobus, pero un autobus puede ser conducido por varios conductores.



Ejercicio 2. Pasa a tablas el siguiente esquema entidad relación (1,5 puntos)



Prestador (**idPrestador**, nombre, disponible)

Deudor (**idDeudor**, nombre, disponible)

Prestamo (**idPrestamo**, importe, interes, finalidad, idDeudor)

Aporta (idPrestador, idPrestamo, cantidad)

Tablas	Claves primarias
Prestador	idPrestador
Deudor	idDeudor
Prestamo	idPrestamo
Aporta	idPrestador, idPrestamo
Tablas	Claves externas
Prestamo	idDeudor a la tabla Deudor.
Aporta	idPrestador a la tabla Prestador idPrestamo a la tabla Prestamo

Ejercicio 3. Disponemos de la siguiente Base de Datos para gestionar la información de los pubs de una determinada provincia. (2,5 puntos)

- 1,5 creación de tablas.
- 1 definición de la restricciones de referencial, valor,...

PUB	TITULAR	EMPLEADO
idPub	IdTitular	idEmpleado
Nombre	Nombre	Nombre
Licencia_fiscal	Domicilio	Domicilio
Domicilio	DNI_Titular	DNI_Empleado
FechaApertura	IdPub	
Horario		
idLocalidad		
EXISTENCIAS	LOCALIDAD	PUB_EMPLEADO
IdArticulo	idLocalidad	idPub
Nombre	Nombre	idEmpleado
Cantidad		funcion
Precio		
idPub		

Se pide escribir los comandos SQL que permitan la creación de las tablas anteriores teniendo en cuenta las siguientes restricciones:

- Todos los valores son de tipo carácter excepto los campos FECHAAPERTURA (fecha) , CANTIDAD, PRECIO y IDLOCALIDAD (numéricos).
- Los únicos campos que no son obligatorios son los campos DOMICILIO.
- Los valores del campo horario sólo pueden ser HOR1, HOR2 y HOR3.
- No es posible dar de alta EXISTENCIAS a precio 0.
- El campo función de la tabla PUB_EMPLEADO sólo puede tener los valores CAMARERO, SEGURIDAD, LIMPIEZA.
- Se ha de mantener la integridad referencial entre las tablas, estas restricciones se pueden deducir de las tablas.
- Las claves primarias vienen marcadas están en negritas.
- Las restricciones de integridad, referencial y de valor tienen que tener un nombre.
- Las restricciones de integridad de clave primaria deben escribirse en la sentencias de creación de tablas.
- Las restricciones de integridad de clave externa deben realizarse aparte de la sentencias de creación de la tabla, al igual, que las restricciones de valor.

```
CREATE TABLE pub (
    idPub          VARCHAR(5),
    Nombre         VARCHAR(60)    NOT NULL,
    licencia_fiscal VARCHAR(60)    NOT NULL,
    domicilio      VARCHAR(60),
    fecha_apertura DATE           NOT NULL,
    horario        VARCHAR(60)    NOT NULL,
    idLocalidad    INTEGER        NOT NULL,
    CONSTRAINT pk_puk PRIMARY KEY (idPub)
);

CREATE TABLE titular (
    idTitular      VARCHAR(5),
    dni_titular    VARCHAR(8)     NOT NULL,
    nombre        VARCHAR(60)     NOT NULL,
    domicilio      VARCHAR(60),
    idPub          VARCHAR(5)     NOT NULL,
    CONSTRAINT pk_Titular PRIMARY KEY (idTitular)
);

CREATE TABLE empleado (
    idEmpleado     VARCHAR(5),
    dni_empleado   VARCHAR(9)     NOT NULL,
    nombre        VARCHAR(60)     NOT NULL,
    domicilio      VARCHAR(60),
    CONSTRAINT pk_empleado PRIMARY KEY (idEmpleado)
);

CREATE TABLE existencias (
    idArticulo     VARCHAR(10),
    Nombre         VARCHAR(60)    NOT NULL,
    Cantidad       INTEGER        NOT NULL,
    Precio         DECIMAL        NOT NULL,
    idPub          VARCHAR(5)     NOT NULL,
    CONSTRAINT pk_existencias PRIMARY KEY (idArticulo)
);

CREATE TABLE localidad (
    Idlocalidad    INTEGER,
    nombre        VARCHAR(60)    NOT NULL,
    CONSTRAINT pk_localidad PRIMARY KEY (idLocalidad)
);

CREATE TABLE pub_empleado (
    idPub          VARCHAR(5),
    idEmpleado     VARCHAR(8),
    funcion        VARCHAR(9),
    CONSTRAINT pk_pub_empleado PRIMARY KEY(idPub,idEmpleado)
);
```

```
ALTER TABLE pub ADD CONSTRAINT fk_pub_localidad FOREIGN KEY
(idLocalidad) REFERENCES localidad(idLocalidad);

ALTER TABLE titular ADD CONSTRAINT fk_titular_pub FOREIGN
KEY (idPub) REFERENCES pub(idPub);

ALTER TABLE existencias ADD CONSTRAINT fk_existencias_pub
FOREIGN KEY (idPub) REFERENCES pub(idPub);

ALTER TABLE pub_empleado ADD CONSTRAINT fk_pubemple_pub
FOREIGN KEY (idPub) REFERENCES pub(idPub);

ALTER TABLE pub_empleado ADD CONSTRAINT fk_pubemple_empleado
FOREIGN KEY (idEmpleado) REFERENCES empleado
(idEmpleado);

ALTER TABLE pub ADD CONSTRAINT ck_horario CHECK (horario IN
('HOR1', 'HOR2', 'HOR3'));

ALTER TABLE existencias ADD CONSTRAINT ck_precio CHECK
(precio <> 0);

ALTER TABLE pub_empleado ADD CONSTRAINT ck_funcion CHECK
(funcion IN ('CAMARERO', 'SEGURIDAD', 'LIMPIEZA'));
```

Ejercicio 4. Sobre la base de datos jardinería, realiza las siguientes consultas:

Consulta 1. Selecciona el código y nombre de producto de mayor precio de venta. (0,5 puntos)

```
SELECT codigo_producto, nombre
FROM producto
WHERE precio_venta =
(SELECT MAX(precio_venta) FROM producto);
```

Consulta 2. Selecciona el código, el nombre y las unidades de los productos, que más unidades tiene en stock y que menos unidades tiene en stock. Se deben de dos mostrar al menos dos productos. (1 punto)

```
SELECT codigo_producto, nombre, cantidad_en_stock
FROM producto
WHERE cantidad_en_stock =
(SELECT max(cantidad_en_stock) FROM producto)
OR cantidad_en_stock =
(SELECT min(cantidad_en_stock) FROM producto);
```

Consulta 3. Obtener para cada pedido: el código de pedido, fecha de pedido, nombre del cliente y el importe total de ese pedido. (1,5 puntos)

Opción 1.

```
SELECT codigo_pedido, fecha_pedido,
nombre_cliente, sum(precio_unidad*cantidad)
FROM DETALLE_PEDIDO DP, PEDIDO P, CLIENTE C
WHERE DP.codigo_pedido = P.codigo_pedido
AND C.codigo_cliente = P.codigo_cliente
GROUP BY codigo_pedido;
```

Opción 2.

```
SELECT P.codigo_pedido, fecha_pedido, nombre_cliente,
SUM(precio_unidad*cantidad)
FROM detalle_pedido DP INNER JOIN pedido P
ON DP.codigo_pedido = P.codigo_pedido
INNER JOIN cliente C
ON P.codigo_cliente = C.codigo_cliente
GROUP BY P.codigo_pedido, fecha_pedido, nombre_cliente;
```

Ejercicio 5. Normalización (1punto)

Dada la siguiente tabla:

idOrden	Fecha	idCliente	NomCliente	Ciudad	idArt	nomArt	Cant	Precio
2301	23/02/22	101	Antonio	Estella	2786	Red	3	35,00
2301	23/02/22	101	Antonio	Estella	4011	Raqueta	6	65,00
2301	23/02/22	101	Antonio	Estella	9132	Paq-3	8	4.75
2302	24/02/22	108	Paco	Pamplona	5794	Paq-6	4	5.00
2303	25/02/22	109	Pedro	Logroño	4011	Raqueta	2	65.00
2303	25/02/22	109	Pedro	Logroño	3141	Funda	2	10.00

Normalizar hasta la tercera forma normal.

Lo primero que tenemos que hacer es determinar la clave primaria, en este caso, se puede observar que la clave primaria es (idOrden, idArt).

PRIMERA FORMA NORMAL.

Al ser una tabla ya está en primera forma normal, ya que los atributos no pueden tomar valores no atómicos, es decir, un atributo no puede tomar varios valores.

SEGUNDA FORMA NORMAL

La segunda forma normal dice que los atributos deben depender completamente de la clave, Vamos a estudiar atributo por atributos.

- Fecha → Depende solamente de idOrden, no está relacionado con idArt.
- idCliente → Depende solamente de idOrden, no está relacionado con idArt.
- NomCliente → Depende solamente de idOrden, no está relacionado con idArt.
- Ciudad → Depende solamente de idOrden, no está relacionado con idArt.
- nomArt → Depende solamente de idArt.
- Cant y Precio → depende de ambos.

Luego, podemos decir que existen las siguientes dependencias:

- idOrden → Fecha, idCliente, nomCliente, ciudad
- idArt → nomArt
- idOrden, idArt → Cant, Precio

Luego, se puede no está en segunda normal. Luego, vamos a generar la tablas orden y articulo.

- **Tabla:** Orden(idOrden, Fecha, idCliente, nomCliente, Ciudad)
- **Tabla:** Articulo(idArt, nomArt)
- **Tabla:** TablaOrigen(idOrden, idArt, cant, precio)

Luego, los datos quedaría de la siguiente forma distribuidos

Tabla: Orden

idOrden	Fecha	idCliente	NomCliente	Ciudad
2301	23/02/22	101	Antonio	Estella
2302	24/02/22	108	Paco	Pamplona
2303	25/02/22	109	Pedro	Logroño

Tabla: Artículo

idArt	nomArt
2786	Red
9132	Paq-3
5794	Paq-6
4011	Raqueta
3141	Funda

Tabla: TablaOrigen

idOrden	idArt	Cant	Precio
2301	2786	3	35,00
2301	4011	6	65,00
2301	9132	8	4.75
2302	5794	4	5.00
2303	4011	2	65.00
2303	3141	2	10.00

TERCERA FORMA NORMAL.

Tanto la tabla articulo y tabla origen está en tercera forma normal.

- **Tabla articulo.** El atributo nomArt no puede implicar ninguno, ya que no hay más atributos.
- **TablaOrigen:** Está en tercera forma normal, ya que no existe dependencias entre los atributos cant y precio.
- **Tabla Orden:**

Vamos a estudiar las dependencias si existen dependencias entre los atributos no clave.

- **Fecha** → Depende de idOrden.
 - **Idcliente** → Depende de idOrden.
 - **NomCliente** → Depende de idOrden y además de idCliente.
 - **Ciudad** → Depende de idOrden y además de idCliente.
- Luego, se pueden establecer las siguientes dependencias:
- Idcliente → NomCliente
 - IdCliente → Ciudad.

Luego, la tabla orden se va a dividir en dos tablas:

- Tabla: Orden_1 (idOrden, fecha, idCliente).
- Tabla: Cliente (idCliente, nomCliente, Ciudad)

Luego, los datos quedarían distribuidos de la siguiente manera:

Tabla: Orden_1

idOrden	Fecha	idCliente
2301	23/02/22	101
2302	24/02/22	108
2303	25/02/22	109

Tabla: Cliente

idCliente	NomCliente	Ciudad
101	Antonio	Estella
108	Paco	Pamplona
109	Pedro	Logroño

Luego, la solución sería:

- Tabla: Orden_1 (**idOrden**, fecha, idCliente).
- Tabla: Cliente (**idCliente**, nomCliente, Ciudad)
- Tabla: Artículo(**idArt**, nomArt)
- Tabla: TablaOrigen(**idOrden**, idArt, cant, precio)

Luego, los datos finalmente se quedarían distribuidos en las cuatro tablas de la siguiente forma:

Tabla: Orden_1

idOrden	Fecha	idCliente
2301	23/02/22	101
2302	24/02/22	108
2303	25/02/22	109

Tabla: Cliente

idCliente	NomCliente	Ciudad
101	Antonio	Estella
108	Paco	Pamplona
109	Pedro	Logroño

Tabla: Artículo

idArt	nomArt
2786	Red
9132	Paq-3
5794	Paq-6
4011	Raqueta
3141	Funda

Tabla: TablaOrigen

idOrden	idArt	Cant	Precio
2301	2786	3	35,00
2301	4011	6	65,00
2301	9132	8	4.75
2302	5794	4	5.00
2303	4011	2	65.00
2303	3141	2	10.00