

APELLIDOS: _____

OPCIÓN: B

NOMBRE: _____

FECHA: 26/01/23

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.

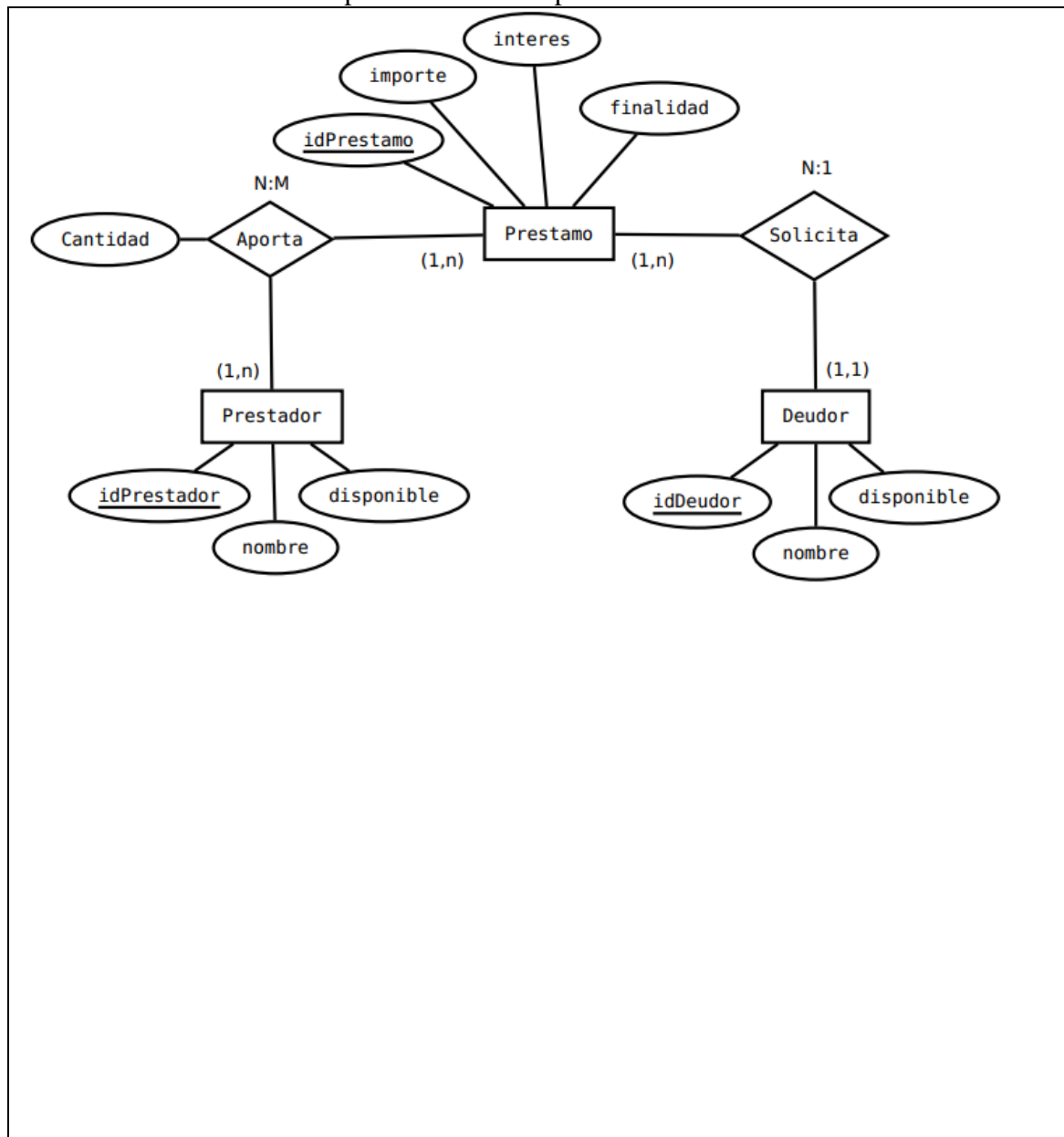
Verificado por el profesor:

SUERTE

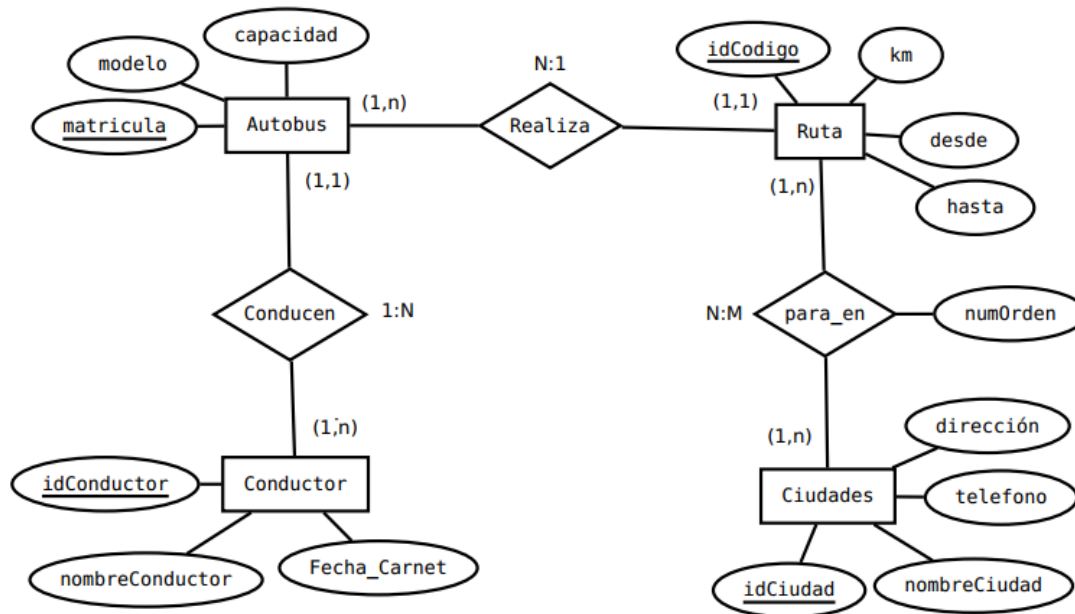
Ejercicio 1. Construye el esquema entidad relación. (2 puntos)

Prestame.com es una empresa que se dedica al préstamo entre personas.

- Los prestadores que desean prestar dinero a otros se registran con un id, su nombre y la cantidad de dinero disponible para las operaciones. Los deudores se identifican por su id y, además, el sistema almacena su nombre y un valor de riesgo en función de su situación personal.
- Cuando el deudor solicita un préstamo, se añade un código de préstamo, el importe total, el plazo de devolución, su interés y la finalidad del mismo. Los prestadores indican qué cantidad quieren aportar a un préstamo. Un prestador puede aportar distintas cantidades parciales a varios préstamos.



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



Ciudades (**idCiudad**, nombreCiudad, Telefono, Dirección)

Ruta(**idCodigo**, km, desde, hasta)

Ruta_Ciudades(**idCodigoRuta,idCuidad**, numOrden)

Autobus (**matricula**, modelo, capacidad, **idCodigoRuta**)

Conductor (**idConductor**, nombreConductor, Fecha_Carnet, **matricula**)

Tablas	Claves Primarias
Ciudades	idCiudad
Ruta	idCodigo
Rutas_Ciudades	idCodigoRuta,idCuidad
Autobus	Matricula
Conductor	idConductor
Tablas	Claves externas
Rutas_Ciudades	idCodigoRuta a la tabla Ruta. idCuidad a la tabla Ciudades
Autobus	idCodigoRuta a la tabla Ruta
Conductor	Matricula a la tabla Autobus

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,...

ESPECTACULOS	PRECIO_ESPECTACULOS	RECINTOS
IdEspectaculo	IdEspectaculo	idRecinto
Nombre	idRecinto	Nombre
Tipo	idZona	Dirección
Fecha_inicial	Precio	Ciudad
Fecha_final		Teléfono
Interperte		horario
idRecinto		
ZONAS_RECINTOS	ASIENTOS	REPRESENTACIONES
idRecinto	idRecinto	IdEspectaculo
idZona	idZona	Fecha
Capacidad	Fila	Horario
	asiento	

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 fechas_inicial, fecha_final son fechas y precio, cantidad, filas (numéricos), mientras que asiento es un carácter, y solo pueden tomar valores de la A la Z.
- Los únicos campos que no son obligatorios son los campos ciudad y dirección de la tabla recintos.
- Los valores del campo horario sólo pueden ser HOR1, HOR2 y HOR3.
- No es posible dar de alta filas que sean menores o iguales a 0.
- Se ha de mantener la integridad referencial entre las tablas, éstas 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 las sentencias de creación de tablas.
- Las restricciones de integridad de clave externa deben realizarse aparte de las sentencias de creación de la tabla, al igual, que las restricciones de valor.

```
ALTER SESSION SET "_ORACLE_SCRIPT"=true;
DROP USER examen CASCADE;
CREATE USER examen IDENTIFIED BY examen DEFAULT TABLESPACE
users;
GRANT CONNECT, RESOURCE,DBA TO examen;
connect examen/examen;

DROP TABLE representaciones CASCADE CONSTRAINTS;
DROP TABLE espectaculos CASCADE CONSTRAINTS;
DROP TABLE recintos CASCADE CONSTRAINTS;
DROP TABLE precio_espectaculos CASCADE CONSTRAINTS;
DROP TABLE zonas_recintos CASCADE CONSTRAINTS;
DROP TABLE asientos CASCADE CONSTRAINTS;

CREATE TABLE espectaculos (
    idEspectaculo VARCHAR2(10),
    nombre VARCHAR2(10) NOT NULL,
    tipo VARCHAR2(10) NOT NULL,
    fechaInicial DATE NOT NULL,
    fechaFinal DATE NOT NULL,
    interprete VARCHAR2(10) NOT NULL,
    idRecinto VARCHAR2(10) NOT NULL,
    CONSTRAINT pk_espectaculo PRIMARY KEY (idEspectaculo)
);

CREATE TABLE recintos (
    idRecinto VARCHAR2(10) NOT NULL,
    nombre VARCHAR2(10) NOT NULL,
    direccion VARCHAR2(10),
    ciudad VARCHAR2(10),
    telefono VARCHAR2(10) NOT NULL,
    horario VARCHAR2(10) NOT NULL,
    CONSTRAINT pk_recintos PRIMARY KEY(idRecinto)
);

CREATE TABLE precio_espectaculos (
    idEspectaculo VARCHAR2(10),
    idRecinto VARCHAR2(10),
    idZona VARCHAR2(10),
    precio NUMBER(5) NOT NULL,
    CONSTRAINT pk_precio_espectaculos PRIMARY KEY
        (idEspectaculo, idRecinto, idZona)
);

CREATE TABLE zonas_recintos (
    idRecinto VARCHAR2(10),
    idZona VARCHAR2(10),
```

```
capacidad NUMBER(5) NOT NULL,
CONSTRAINT pk_zonas_recintos PRIMARY KEY
    (idRecinto, idZona)
);

CREATE TABLE asientos (
    idRecinto VARCHAR2(10),
    idZona VARCHAR2(10),
    fila NUMBER,
    asiento NUMBER,
CONSTRAINT pk_asiento PRIMARY KEY
    (idRecinto, idZona, fila, asiento)
);

CREATE TABLE representaciones (
    idEspectaculo VARCHAR2(10),
    fecha DATE,
    horario VARCHAR2(10),
CONSTRAINT pk_representaciones PRIMARY KEY
    (idEspectaculo, fecha, horario)
);

-- Restricciones de clave externa
-- Tabla Espectaculos
ALTER TABLE espectaculos ADD
    CONSTRAINT fk_espectaculos_idRecinto FOREIGN KEY
        (idRecinto) REFERENCES recintos(idRecinto);

-- Tabla Precio_espectaculos
ALTER TABLE precio_espectaculos ADD (
    CONSTRAINT fk_precios_espectaculos_idEspectaculo
        FOREIGN KEY (idEspectaculo) REFERENCES
            espectaculos(idEspectaculo),
    CONSTRAINT fk_precios_espectaculos_idRecinto_idZona
        FOREIGN KEY (idRecinto, idZona) REFERENCES
            zonas_recintos(idRecinto, idZona));

-- Tabla zonas_recintos
ALTER TABLE zonas_recintos ADD
    ONSTRAINT fk_zonas_recintos_idRecinto FOREIGN KEY
        (idRecinto) REFERENCES recintos(idRecinto);

-- Tabla asientos
ALTER TABLE asientos ADD
    CONSTRAINT fk_asientos_idZona FOREIGN KEY
        (idRecinto, idZona) REFERENCES
```

```
zonas_recintos(idRecinto, idZona);

-- Tabla Representaciones
ALTER TABLE representaciones ADD
    CONSTRAINT fk_representaciones FOREIGN KEY
        (idEspectáculo) REFERENCES
            espectaculos(idEspectaculo);

-- Restricciones de valor
ALTER TABLE precio_espectaculos ADD
    CONSTRAINT check_precio_espectaculos_precio
        CHECK (precio>0);
ALTER TABLE recintos ADD
    CONSTRAINT check_recintos_horario
        CHECK (horario IN ('HOR1', 'HOR2', 'HOR3'));

ALTER TABLE zonas_recintos ADD
    CONSTRAINT check_zonas_recintos_capacidad
        CHECK (capacidad>0);

ALTER TABLE asientos ADD(
    CONSTRAINT check_asientos_fila CHECK (fila>0),
    CONSTRAINT check_asientos_asiento
        CHECK (asiento BETWEEN 'A' and 'Z'));

ALTER TABLE representaciones ADD
    CONSTRAINT check_representaciones_horario
        CHECK (horario IN ('HOR1', 'HOR2', 'HOR3'));
```

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

Consulta 1. Selecciona el código y nombre de producto del cual tenemos más existencias en stock. **(0,5 puntos)**

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

Consulta 2. Selecciona el código, el nombre y precio, del producto más caro y más barato, en el caso de que existan varios deben mostrarse todos. **(1 punto)**

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

Consulta 3. Obtener para cada cliente: el código de cliente, nombre de cliente, el número de pedidos realizados y el importe total de los pagos efectuados. **(1,5 puntos)**

Opción 1.

```
SELECT C.codigo_cliente, C.nombre_cliente,
COUNT(*) AS "Numero Pedidos",
SUM(DP.cantidad*DP.precio_unidad) AS "Importe"
FROM cliente C, pedido P, detalle_pedido DP
WHERE P.codigo_cliente = C.codigo_cliente
AND DP.codigo_pedido = P.codigo_pedido
GROUP BY C.codigo_cliente, C.nombre_cliente;
```

Opción 2.

```
SELECT C.codigo_cliente, C.nombre_cliente,
COUNT(*) AS "Numero Pedidos",
SUM(DP.cantidad*DP.precio_unidad) AS "Importe"
FROM cliente C INNER JOIN pedido P
ON P.codigo_cliente = C.codigo_cliente
INNER JOIN detalle_pedido DP
ON DP.codigo_pedido = P.codigo_pedido
GROUP BY C.codigo_cliente, C.nombre_cliente;
```


Ejercicio 5. Normalización (1punto)

Dada la siguiente tabla:

idOrden	Fecha	idCliente	NomCliente	Cuidad	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