



# Bases de Datos

Gabriel Marín Díaz

# 5

## Actividades

A continuación se presenta la actividad a realizar para evaluar el aprendizaje durante el curso.



# **ACTIVIDAD 1**



# Ejercicio 1

Nos contratan para realizar el diseño funcional de base de datos (diagrama Entidad – Relación y Relacional) de un modelo de negocio de un mayorista de moda que distribuye a 12.000 tiendas en todo el mundo, con las siguientes restricciones:

- Para cada tienda necesitamos almacenar la siguiente información: ID de tienda (único), direcciones de entrega (varias por tienda), límite de crédito de cada tienda (en ningún caso debe superar los 30.000 €) y nivel de descuento que estará en función de la categoría de tienda y en ningún caso superará el 20%.
- Las tiendas se dividen en categorías, de acuerdo con el volumen de compras anuales, de forma que a mayor volumen de compra los descuentos serán mayores.
- Las tiendas realizan pedidos al mayorista, es fundamental identificar en el pedido la tienda que lo realiza, la dirección de entrega, así como la fecha en la que se realiza dicho pedido.
- Cada pedido dispone de líneas de pedido, en cada línea de pedido se determinará el identificador del artículo, descripción, su número de unidades y el importe de este.
- Los artículos se distribuyen a cada tienda por varios proveedores, para identificar a cada proveedor se tendrá en cuenta, el ID de proveedor (único) y dirección de contacto.

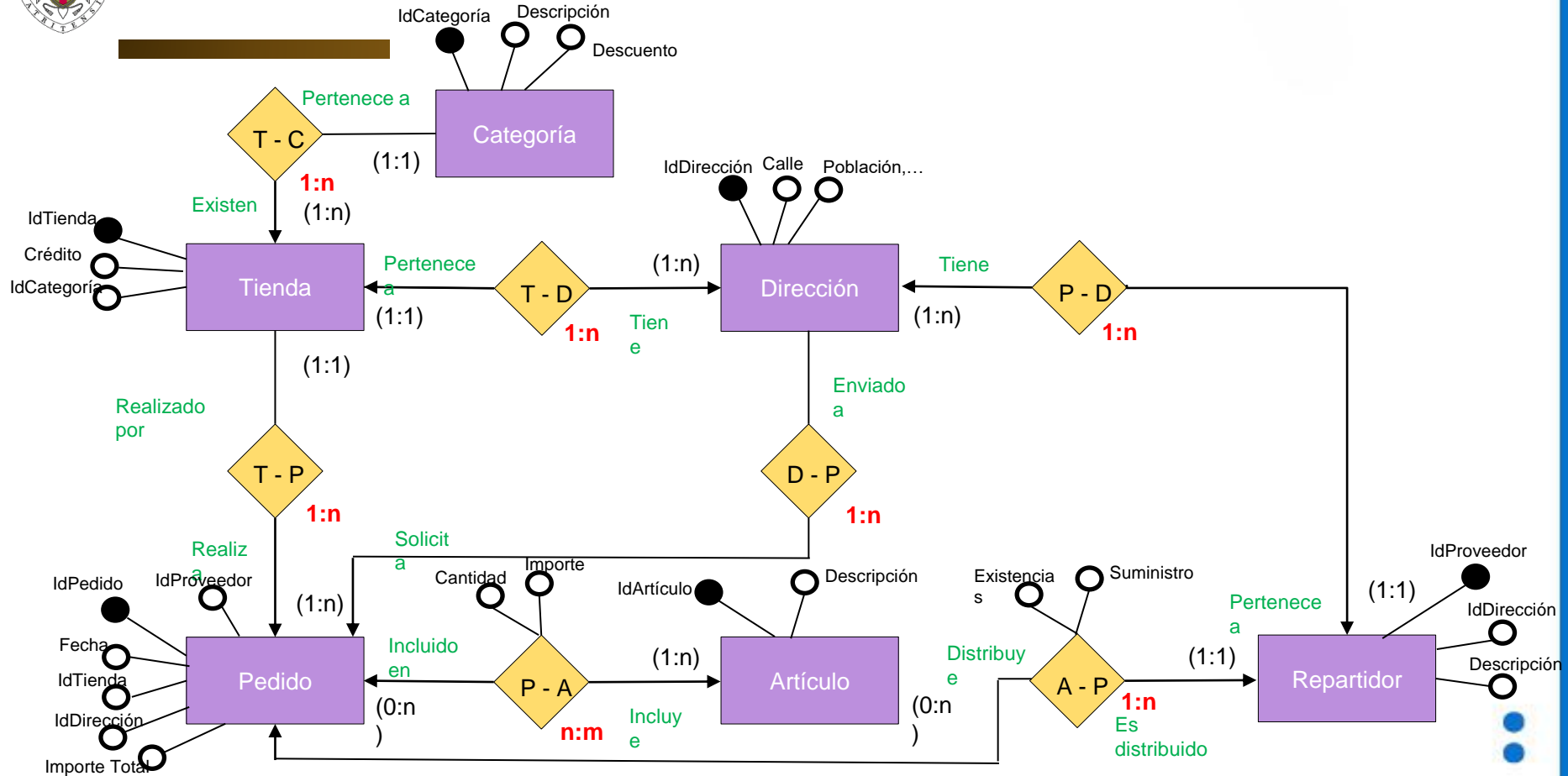
Nota: Una dirección se entenderá como N<sup>o</sup>, Calle, Población, Ciudad, Teléfono y email. Una fecha incluye hora.

Determinar el diagrama Entidad – Relación y el paso a tablas (Modelo Relacional).



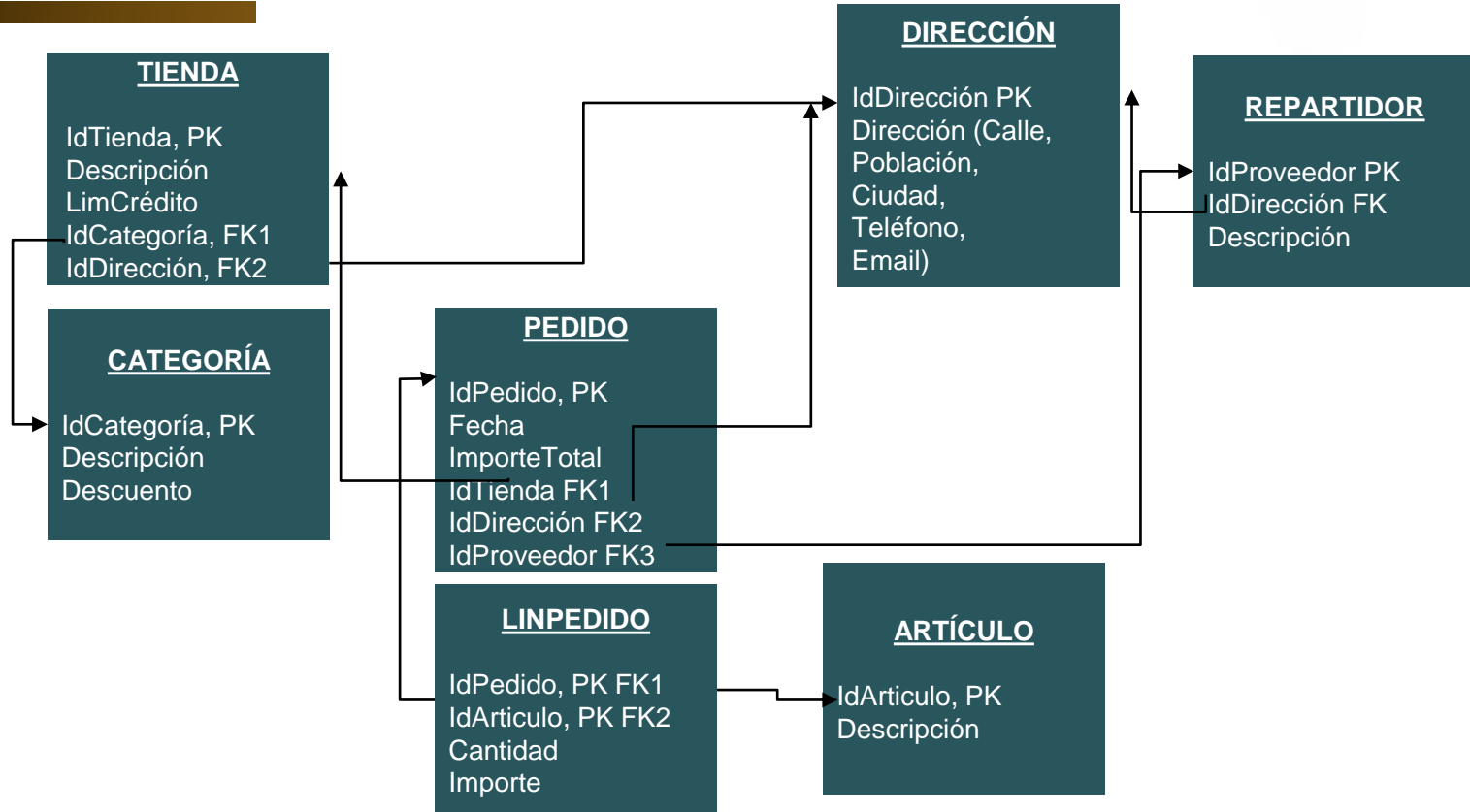


# Ejercicio 1





# Ejercicio 1





## Ejercicio 2

1. Crear mediante instrucciones SQL las entidades de acuerdo con el modelo relacional definido, es fundamental detallar para cada entidad los atributos que forman parte de la PK y las FK, no es necesario entrar en detalle con el resto de los atributos de cada entidad.

### YA DEFINIDO EN EL DIAGRAMA RELACIONAL

2. Utilizar la instrucción SQL de inserción de datos para insertar una fila en la tabla de pedidos.

```
INSERT INTO PEDIDO (IdPedido, Fecha, Importe Total, IdTienda, IdDirección, IdProveedor) VALUES ('0001', '2020/01/01', 500, '001', '001', '001');
```

OJO!!! PARA QUE ESTE INSERT FUNCIONE ES NECESARIO INSERTAR EN LA TABLA DE TIENDAS, DIRECCIONES, Y PROVEEDORES DATOS.

3. Visualizar mediante una instrucción SQL todas las tiendas que componen la red de distribución de la fábrica, se deberán detallar: nombre de la tienda, dirección, descripción de la categoría, descuento y límite de crédito asociado a la tienda.

```
SELECT T.Descripción T, D.Calle, D.Población, D.Ciudad, D.CodigoPostal, D.Teléfono, C.Descripción, T.LimCredito, C.Descuento FROM TIENDA T, DIRECCIÓN D, CATEGORÍA C WHERE T.IdCategoría = C.IdCategoría AND T.IdDirección = D. IdDirección GROUP BY T.IdTienda;
```





## Ejercicio 2

4. Visualizar mediante una instrucción SQL los pedidos suministrados a cada una de las tiendas en un período determinado (último año). Se deberán obtener los siguientes datos: número de pedido, fecha de suministro, dirección de entrega, y el importe total del pedido.

```
SELECT P.IdPedido, P.Fecha, D.Dirección, P.ImporteTotal FROM TIENDA T, PEDIDO P, DIRECCIÓN D WHERE  
T.IdTienda = P.IdTienda AND P.Fecha BETWEEN '2019-01-01' AND '2019-12-31' AND P.IdDirección = D.IdDirección  
GROUP BY T.IdTienda;
```

5. Identificar mediante una consulta SQL los repartos realizados por cada uno de los proveedores destinados a ello. Se deberá identificar al menos: Nombre del proveedor de reparto, su dirección y la relación de los artículos suministrados en cada reparto.

```
SELECT R.Descripción, D.Dirección, LP.IdArticulo, A.Descripción FROM REPARTIDOR R, DIRECCIÓN D, ARTICULO A,  
LINPEDIDO LP, PEDIDO P WHERE R.IdProveedor = P.IdProveedor AND P.IdPedido = LP.IdPedido AND LP.IdArticulo =  
A.IdArticulo AND D.IdDirección = R.IdDirección GROUP BY R.IdProveedor;
```

6. Totalizar los repartos anuales realizados por cada proveedor de reparto.

```
SELECT R.Descripción, SUM(P.ImporteTotal) FROM REPARTIDOR R, PEDIDO P WHERE R.IdProveedor =  
P.IdProveedor AND FECHA BETWEEN '2019-01-01' AND '2019-12-31' GROUP BY R.Descripción;
```





## Ejercicio 2

7. Identificar los cambios a realizar en el modelo relacional y en BBDD para clasificar a los proveedores de reparto en categorías, de la misma forma que clasificamos las tiendas por categorías.
- Será necesario crear una relación entre proveedores de reparto y categorías, de la misma forma que entre tienda y categoría.
  - En BD será necesario crear una relación entre ambas, para ello se realizará un ALTER TABLE de forma que se añada el campo IdCategoría en la tabla REPARTIDOR.
  - En el ALTER TABLE habrá que especificar el FK entre IdCategoría de ambas tablas.

```
ALTER TABLE REPARTIDOR ADD IdCategoría;  
ALTER TABLE REPARTIDOR ADD CONSTRAINT FK_IdCategoría FOREIGN KEY (IdCategoría) REFERENCES  
CATEGORÍA (IdCategoría);
```

8. Necesitamos introducir nuevos atributos en la tabla de artículos, la fábrica ha descubierto que puede comprar un artículo de parecidas características al nuestro y distribuirlo como marca blanca. Necesitaríamos:
- Ampliar la tabla de artículos para diferenciar el artículo que fabricamos, del artículo que compramos a un tercero (proveedor).

```
ALTER TABLE ARTICULO ADD IdFabricante;  
ALTER TABLE ARTICULO MODIFY PRIMARY KEY (IdArticulo, IdFabricante);
```

\*\* NOTA: Si queremos hacerlo mejor, se crearía una entidad de fabricantes y estableceríamos un vínculo entre artículo / fabricante.





## Ejercicio 2

- Ampliar la tabla de artículos para incorporar el coste de este (fabricación o bien el de compra).

```
ALTER TABLE ARTICULO ADD CosteFab;
```

- Identificar los cambios a realizar a nivel lógico (diagrama relacional) y físico (BBDD).

YA EXPLICADOS.

9. Queremos ampliar la información del proveedor de suministro, para ello necesitaríamos incorporar los datos relativos a las zonas de cobertura de este (Países y Regiones). Determinar los cambios a realizar a nivel físico y lógico.

```
CREATE TABLE COBERTURA (IdCobertura VARCHAR(4), País VARCHAR (255), Región VARCHAR (255), PRIMARY KEY IdCobertura);
```

```
ALTER TABLE REPARTIDOR ADD IdCobertura;
```

```
ALTER TABLE REPARTIDOR MODIFY PRIMARY KEY (IdProveedor, IdCobertura);
```

Establecer una relación entre coberturas y proveedores... se ve en el ALTER TABLE.



## Ejercicio 2

10. ¿Qué podríamos hacer para realizar un backup de la tabla de pedidos / líneas de pedido? Esto es, necesitamos hacer todos los días un proceso de backup a otra Base de Datos en las que se consolida toda la venta del grupo (pedidos, líneas de pedido).
- CREAR UNA TABLA DE RESPALDO QUE RECOJA LA INFORMACIÓN DE PEDIDOS Y OTRA TABLA QUE RECOJA LAS DE LÍNEAS DE PEDIDO.
  - CREAR UN TRIGGER DE FORMA QUE CADA VEZ QUE SE REALICE UN APUNTE EN PEDIDO / LINEA DE PEDIDO SE REALICE UNA INSERCIÓN EN LAS TABLAS CREADAS.

```
CREATE TRIGGER TR_INSERTPEDIDO AFTER INSERT ON PEDIDO
FOR EACH ROW
BEGIN
    INSERT INTO BACKUP.PEDIDO (IdPedido, Fecha,...) VALUES (new.IdPedido, new.Fecha,...);
END //
```

HACER LO MISMO PARA LÍNEAS DE PEDIDO.



# Rúbrica

La entrega se realizará mediante una presentación en formato pptx y constará al menos de los siguientes puntos:

- Resumen de los requisitos.
- Diagrama Entidad / Relación.
- Diagrama Relacional.
- Respuesta a las sentencias SQL solicitadas.
- Conclusiones.

Por otro lado, además será necesario entregar el fichero .sql de MySQL.

## **Rúbrica:**

- Diagrama entidad / relación detallado - 2 puntos.
- Diagrama relacional detallado - 2 puntos.
- Fichero .SQL - 5 puntos (0,5 puntos / pregunta acertada).
- Presentación y documentación completa – 1 puntos.



*GRACIAS!!!*