

Módulo Bases de Datos SQL

Alumno: Santiago Torres Busquets



UNIVERSIDAD
COMPLUTENSE
MADRID



RESUMEN DE REQUISITOS

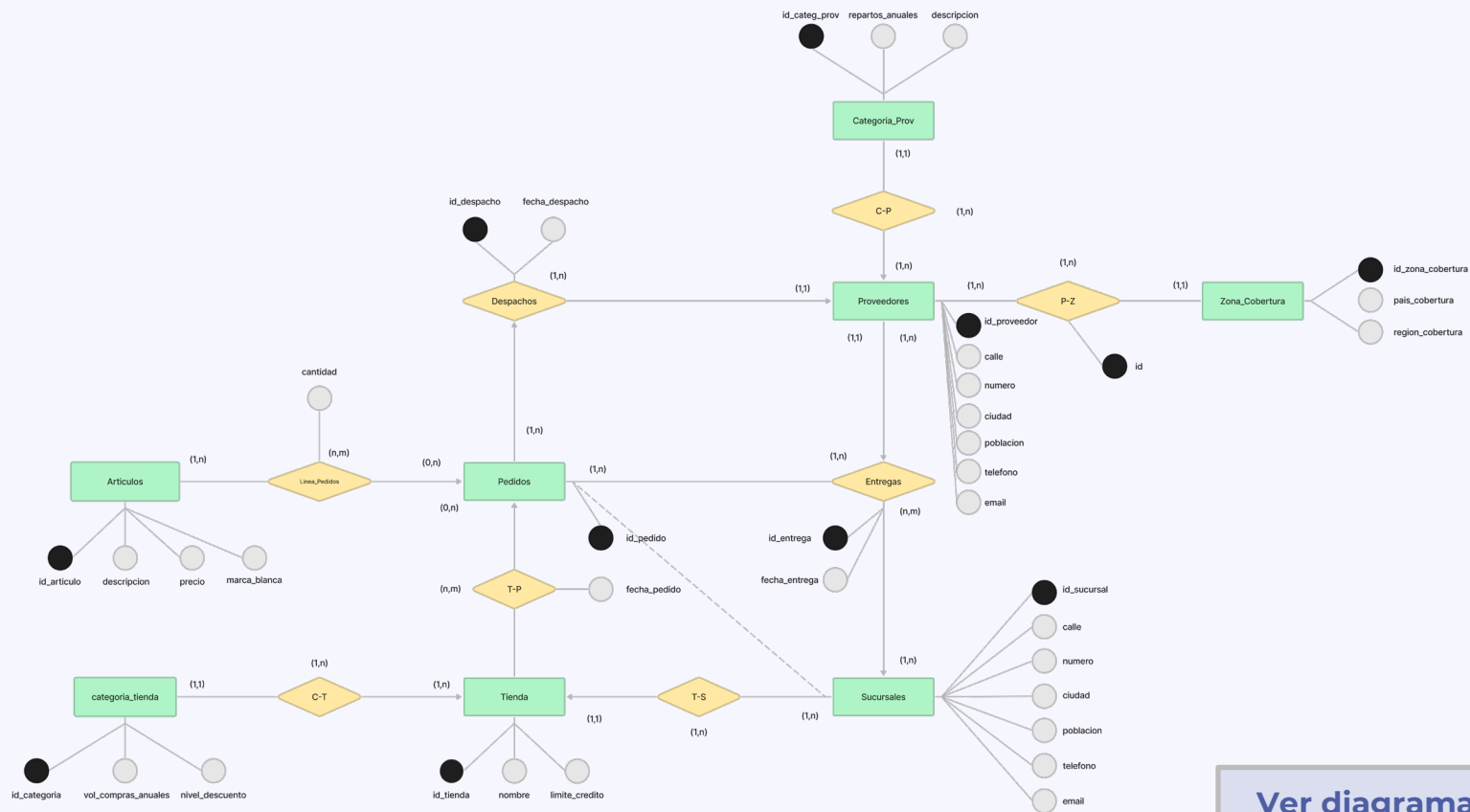
Diseño funcional de BBDD para mayorista que comercializa a tiendas de moda en el mundo:

- Diagrama E-R
- Modelo Relacional
- Consultas a base de datos:
 - Todas las tiendas de la red de distribución
 - Pedidos suministrados a tiendas en el último año
 - Artículos entregados
 - Repartos anuales por proveedor
 - Actualizar modelo relación para categorizar a proveedores.
 - Introducir artículo de marca blanca
 - Incorporar zonas de cobertura de proveedores
 - Backup a otra BBDD de los pedidos y líneas de pedido diarios

01

DIAGRAMA ENTIDAD- RELACION

01. DIAGRAMA ENTIDAD-RELACION



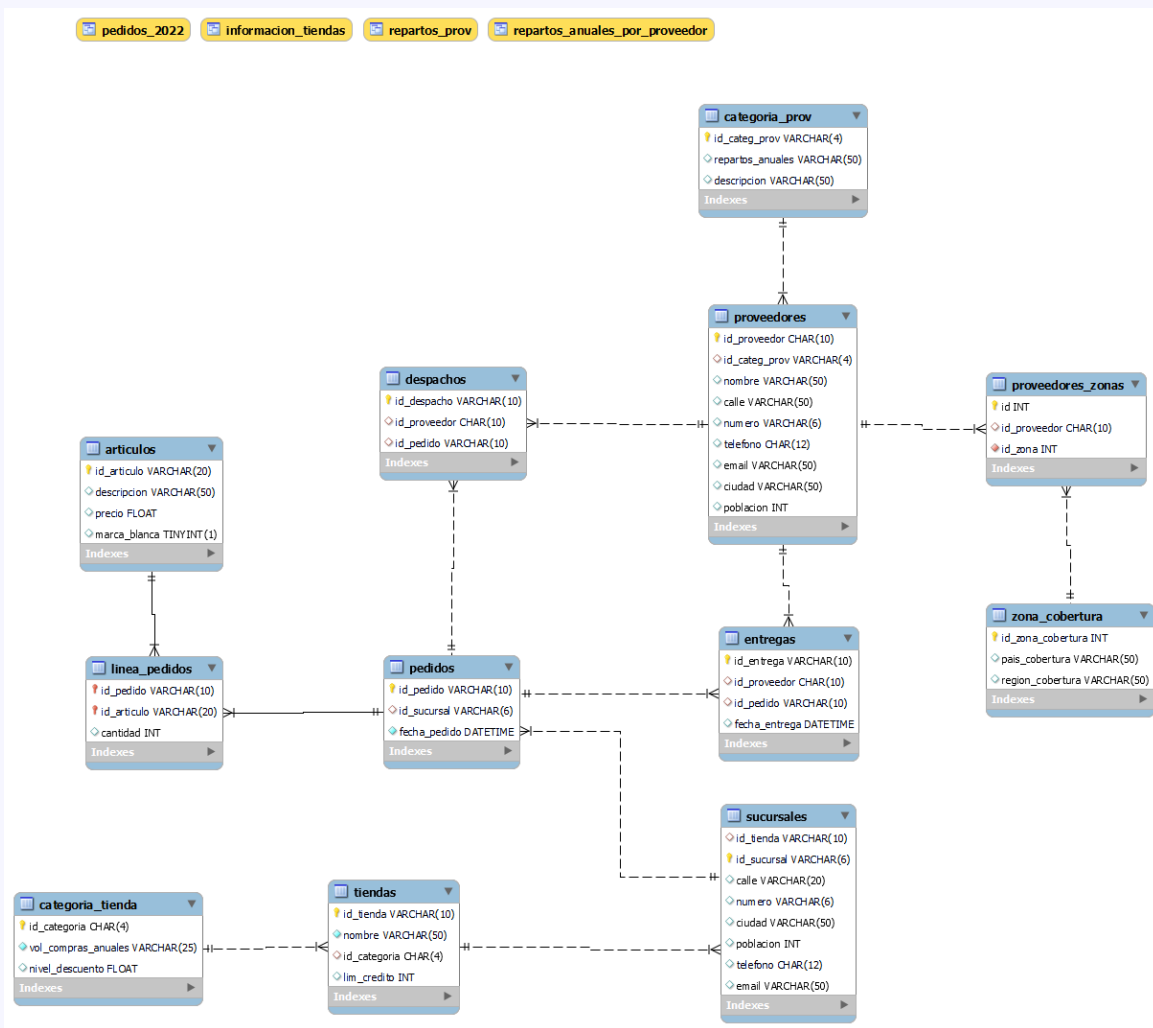
[Ver diagrama online](#)



02

MODELO RELACIONAL

02. MODELO RELACIONAL



03

CONSULTAS SQL

CONSULTAS SQL – nº1

1) Creacion de BD, tablas y definicion de PK y FK

```
DROP DATABASE mayorista;  
CREATE DATABASE mayorista;  
USE mayorista;
```

```
/* La tabla proveedores contiene las compañías de distribución  
de nuestros productos a las tiendas. */
```

```
CREATE TABLE proveedores (  
    id_proveedor CHAR(10),  
    nombre VARCHAR(50),  
    calle VARCHAR(50),  
    numero VARCHAR(6),  
    telefono CHAR(12),  
    email VARCHAR(50),  
    ciudad VARCHAR(50),  
    poblacion INT,  
    PRIMARY KEY (id_proveedor)  
);
```

```
/* La tabla categoria_tienda contiene las categorías en base al volumen de compras anuales,  
que a su vez determinan el nivel de descuento de cada tienda.*/
```

```
CREATE TABLE categoria_tienda (  
    id_categoria CHAR(4),  
    vol_compras_anuales VARCHAR(25) NOT NULL,  
    nivel_descuento FLOAT,  
    PRIMARY KEY (id_categoria),  
    CHECK (nivel_descuento <= 0.2)  
);
```

```
/* La tabla articulos contiene los articulos para la venta. */
```

```
CREATE TABLE articulos (  
    id_articulo VARCHAR(20),  
    descripcion VARCHAR(50),  
    precio FLOAT,  
    PRIMARY KEY (id_articulo)  
);
```


CONSULTAS SQL – nº1

/* La tabla tiendas almacena el nombre de la tienda y su límite de crédito*/

```
CREATE TABLE tiendas (  
    id_tienda VARCHAR(10),  
    nombre VARCHAR(50) NOT NULL,  
    id_categoria CHAR(4),  
    lim_credito INT,  
    PRIMARY KEY (id_tienda),  
    FOREIGN KEY (id_categoria) REFERENCES categoria_tienda(id_categoria),  
    CHECK (lim_credito <= 30000)  
);
```

/* La tabla pedidos acumula los pedidos y su información relevante. */

```
CREATE TABLE pedidos (  
    id_pedido VARCHAR(10),  
    id_sucursal VARCHAR(6),  
    fecha_pedido DATETIME NOT NULL,  
    PRIMARY KEY (id_pedido),  
    FOREIGN KEY (id_sucursal) REFERENCES sucursales(id_sucursal)  
);
```

/* La tabla sucursales contiene la información básica de cada sucursal,
en base a las tiendas creadas en la tabla tiendas,
dada la posibilidad de que cada tienda tenga más de una sucursal */

```
CREATE TABLE sucursales (  
    id_tienda VARCHAR(10),  
    id_sucursal VARCHAR(6),  
    calle VARCHAR(20),  
    numero VARCHAR(6),  
    ciudad VARCHAR(50),  
    poblacion INT,  
    telefono CHAR(12),  
    email VARCHAR(50),  
    PRIMARY KEY (id_sucursal),  
    FOREIGN KEY (id_tienda) REFERENCES tiendas(id_tienda)
```

CONSULTAS SQL – nº1

/* La tabla linea_pedidos relaciona los pedidos con los articulos. */

```
CREATE TABLE linea_pedidos (  
    id_pedido VARCHAR(10),  
    id_articulo VARCHAR(20),  
    cantidad INT,  
    PRIMARY KEY (id_pedido,id_articulo),  
    FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido),  
    FOREIGN KEY (id_articulo) REFERENCES articulos(id_articulo)  
);
```

/* La tabla despachos registra los despachos del mayorista al proveedor para su posterior entrega a la tienda. */

```
CREATE TABLE despachos (  
    id_despacho VARCHAR(10),  
    id_proveedor CHAR(10),  
    id_pedido VARCHAR(10),  
    PRIMARY KEY (id_despacho),  
    FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor),  
    FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido)  
);
```

/* La tabla entregas registra el reparto del pedido a las tiendas. */

```
CREATE TABLE entregas (  
    id_entrega VARCHAR(10),  
    id_proveedor CHAR(10),  
    id_pedido VARCHAR(10),  
    fecha_entrega DATETIME,  
    PRIMARY KEY (id_entrega),  
    FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor),  
    FOREIGN KEY (id_pedido) REFERENCES pedidos(id_pedido)  
);
```

CONSULTAS SQL – nº2

2) Inserción de datos en las tablas.

```
INSERT INTO categoria_tienda (id_categoria, vol_compras_anuales, nivel_descuento) VALUES ('ESP1', 'Hasta 10000 €', 0.05);
INSERT INTO categoria_tienda (id_categoria, vol_compras_anuales, nivel_descuento) VALUES ('ESP2', 'Hasta 50000 €', 0.10);
INSERT INTO categoria_tienda (id_categoria, vol_compras_anuales, nivel_descuento) VALUES ('ESP3', 'Hasta 100000 €', 0.15);
INSERT INTO categoria_tienda (id_categoria, vol_compras_anuales, nivel_descuento) VALUES ('ESP4', 'Mas de 100000 €', 0.19);
```

```
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('001001001', 'Camisa Masculina Negra', 30.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('001001002', 'Camisa Masculina Blanca', 30.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('001002001', 'Camiseta Femenina Negra', 35.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('001002002', 'Camiseta Femenina Blanca', 35.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('002001001', 'Pantalon Masculino Negro', 42.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('002001003', 'Pantalon Masculino Azul', 42.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('002002001', 'Pantalon Femenino Negro', 30.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('002002002', 'Pantalon Femenino Blanco', 30.0);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('003003001', 'Calcetines unisex negro', 8.5);
INSERT INTO articulos (id_articulo, descripcion, precio) VALUES ('003003002', 'Calcetines unisex blanco', 8.5);
```

CONSULTAS SQL – nº2

```
INSERT INTO proveedores (id_proveedor,nombre,calle,numero,telefono,email,ciudad,poblacion)
VALUES ('UE&ESP&001','CORREOS','AV. DE AMERICA','5','+34612123456','delivery@correos.es','madrid',3300000);
INSERT INTO proveedores (id_proveedor,nombre,calle,numero,telefono,email,ciudad,poblacion)
VALUES ('UE&ESP&002','UPS','C. DE LOPEZ DE HOYOS','64','+34600011111','delivery@ups.es','madrid',3300000);
INSERT INTO proveedores (id_proveedor,nombre,calle,numero,telefono,email,ciudad,poblacion)
VALUES ('UE&ESP&003','DHL','C. DEL GENERAL PARDINAS','108','+34612012012','delivery@dhl.es','madrid',3300000);
INSERT INTO proveedores (id_proveedor,nombre,calle,numero,telefono,email,ciudad,poblacion)
VALUES ('UE&ESP&004','AMAZON','C. SIN NOMBRE','1','+34612333333','delivery@amazon.es','madrid',3300000);
```

```
INSERT INTO tiendas (id_tienda, nombre, id_categoria, lim_credito) VALUES ('000001','El Corte Ingles','ESP4',29000);
INSERT INTO tiendas (id_tienda, nombre, id_categoria, lim_credito) VALUES ('000002','Primark','ESP3',25000);
INSERT INTO tiendas (id_tienda, nombre, id_categoria, lim_credito) VALUES ('000003','Zara','ESP2',22000);
INSERT INTO tiendas (id_tienda, nombre, id_categoria, lim_credito) VALUES ('000004','Decathlon','ESP1',15000);
```

CONSULTAS SQL – nº2

```
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000001', '00001', 'C. de Serrano', '65', 'madrid', '3300000', '+34910111222', 'eci1@eci.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000001', '00002', 'C. de Goya', '87', 'madrid', '3300000', '+34910111333', 'eci2@eci.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000001', '00003', 'C. de la Princesa', '56', 'madrid', '3300000', '+34910111444', 'eci3@eci.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000002', '00004', 'C. Gran Via', '32', 'madrid', '3300000', '+34910222333', 'prim@ark.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000002', '00005', 'C. de Aracne', '125', 'madrid', '3300000', '+34910333444', 'pri@mark.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000003', '00006', 'C. Gran Via', '34', 'madrid', '3300000', '+34910444555', 'z@ra.es');
INSERT INTO sucursales (id_tienda, id_sucursal, calle, numero, ciudad, poblacion, telefono, email)
VALUES ('000004', '00007', 'P. de la Castellana', '10', 'madrid', '3300000', '+34910444777', 'dec@thlon.es');
```

```
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (1, '00005', '2021-08-20 20:12:58');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (2, '00004', '2021-07-18 15:38:26');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (3, '00001', '2021-07-07 10:19:01');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (4, '00003', '2022-09-10 17:13:49');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (5, '00003', '2022-04-15 11:01:41');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (6, '00002', '2022-05-28 18:19:22');
INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (7, '00006', '2022-06-30 17:59:59');
```

CONSULTAS SQL – nº2

```
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (1,'001001002',60);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (1,'002001001',10);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (3,'003003002',150);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (2,'001001002',100);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (4,'001001002',30);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (5,'003003001',70);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (6,'002002002',55);
INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (7,'001002001',45);
```

```
INSERT INTO despachos (id_despacho, id_proveedor, id_pedido) VALUES ('AAA000','UE&ESP&002', 2);
INSERT INTO despachos (id_despacho, id_proveedor, id_pedido) VALUES ('AAA001','UE&ESP&004', 3);
```

```
INSERT INTO entregas (id_entrega, id_proveedor, id_pedido, fecha_entrega) VALUES ('ENT001','UE&ESP&002',2,'2021-08-27 12:05:25');
INSERT INTO entregas (id_entrega, id_proveedor, id_pedido, fecha_entrega) VALUES ('ENT002','UE&ESP&004',3,'2021-07-21 16:52:02');
INSERT INTO entregas (id_entrega, id_proveedor, id_pedido, fecha_entrega) VALUES ('ENT003','UE&ESP&004',5,'2022-05-14 16:52:02');
INSERT INTO entregas (id_entrega, id_proveedor, id_pedido, fecha_entrega) VALUES ('ENT004','UE&ESP&001',7,'2022-07-14 16:52:02');
INSERT INTO entregas (id_entrega, id_proveedor, id_pedido, fecha_entrega) VALUES ('ENT005','UE&ESP&001',6,'2022-06-20 16:52:02');
```

CONSULTAS SQL – nº3

Completada la inserción de datos. El siguiente paso aplicar consultas a la BD para corroborar su funcionamiento.
3) Query que detalla todas las tiendas, nombre, direccion, descripcion de categorias, descuento y limite de credito.

```
CREATE VIEW informacion_tiendas AS
SELECT td.nombre as 'Tienda', ct.vol_compras_anuales as 'Vol. Compras Anuales', round(ct.nivel_descuento * 100) as 'Desc. %',
td.lim_credito as 'Limite Credito €', suc.calle, suc.numero, suc.telefono, suc.email, suc.ciudad, suc.poblacion
FROM tiendas AS td
JOIN categoria_tienda AS ct
ON td.id_categoria = ct.id_categoria
JOIN sucursales AS suc
ON td.id_tienda = suc.id_tienda;

SELECT * FROM informacion_tiendas;
```

Tienda	Vol. Compras Anuales	Desc. %	Limite Credito €	calle	numero	telefono	email	ciudad	poblacion
El Corte Ingles	Mas de 100000 €	19	29000	C. de Serrano	65	+34910111222	eci1@eci.es	madrid	3300000
El Corte Ingles	Mas de 100000 €	19	29000	C. de Goya	87	+34910111333	eci2@eci.es	madrid	3300000
El Corte Ingles	Mas de 100000 €	19	29000	C. de la Princesa	56	+34910111444	eci3@eci.es	madrid	3300000
Primark	Hasta 100000 €	15	25000	C. Gran Vía	32	+34910222333	prim@ark.es	madrid	3300000
Primark	Hasta 100000 €	15	25000	C. de Alcalá	125	+34910333444	pri@mark.es	madrid	3300000
Zara	Hasta 50000 €	10	22000	C. Gran Vía	34	+34910444555	z@ra.es	madrid	3300000
Decathlon	Hasta 10000 €	5	15000	P. de la Castellana	10	+34910444777	dec@thlon.es	madrid	3300000

CONSULTAS SQL – nº4

4) Query de pedidos de todas las tiendas en el último año, mostrando num. de pedido, fecha, direccion de entrega e importe total del pedido.

```
CREATE VIEW pedidos_2022 AS
SELECT pd.id_pedido as 'Nº Pedido', td.nombre as 'Tienda', pd.fecha_pedido as 'Fecha de Pedido',
suc.calle, suc.numero, lp.cantidad, ar.precio, (ar.precio*lp.cantidad) as 'importe €'
FROM pedidos as pd
JOIN sucursales as suc
ON pd.id_sucursal=suc.id_sucursal
JOIN tiendas as td
ON suc.id_tienda=td.id_tienda
JOIN linea_pedidos as lp
ON pd.id_pedido=lp.id_pedido
JOIN articulos as ar
ON lp.id_articulo=ar.id_articulo
WHERE year(pd.fecha_pedido) = 2022
ORDER BY pd.id_pedido;

SELECT * FROM pedidos_2022;
```

	Nº Pedido	Tienda	Fecha de Pedido	calle	numero	cantidad	precio	importe €
▶	4	El Corte Ingles	2022-09-10 17:13:49	C. de la Princesa	56	30	30	900
	5	El Corte Ingles	2022-04-15 11:01:41	C. de la Princesa	56	70	8.5	595
	6	El Corte Ingles	2022-05-28 18:19:22	C. de Goya	87	55	30	1650
	7	Zara	2022-06-30 17:59:59	C. Gran Via	34	45	35	1575

CONSULTAS SQL – nº5

5) Query de repartos realizados por los proveedores identificando nombre del proveedor, dirección y los artículos suministrados en cada entrega;

```
CREATE VIEW repartos_prov AS
SELECT en.id_entrega, en.fecha_entrega as fecha, pv.nombre as proveedor,
pv.calle, pv.numero, en.id_pedido, ar.descripcion, lp.cantidad
FROM entregas AS en
JOIN proveedores AS pv
ON en.id_proveedor=pv.id_proveedor
JOIN linea_pedidos AS lp
ON en.id_pedido=lp.id_pedido
JOIN articulos AS ar
ON lp.id_articulo=ar.id_articulo;

SELECT * FROM repartos_prov;
```

id_entrega	fecha	proveedor	calle	numero	id_pedido	descripcion	cantidad
ENT004	2022-07-14 16:52:02	CORREOS	AV. DE AMERICA	5	7	Camiseta Femenina Negra	45
ENT005	2022-06-20 16:52:02	CORREOS	AV. DE AMERICA	5	6	Pantalon Femenino Blanco	55
ENT001	2021-08-27 12:05:25	UPS	C. DE LOPEZ DE HOYOS	64	2	Camisa Masculina Blanca	100
ENT002	2021-07-21 16:52:02	AMAZON	C. SIN NOMBRE	1	3	Calcetines unisex blanco	150
ENT003	2022-05-14 16:52:02	AMAZON	C. SIN NOMBRE	1	5	Calcetines unisex negro	70

CONSULTAS SQL – n°6

6) Repartos anuales totales por cada proveedor de reparto

```
CREATE VIEW repartos_anuales_por_proveedor AS
SELECT year(en.fecha_entrega) AS 'Año', pv.nombre, COUNT(en.id_entrega) AS 'Cantidad de Pedidos'
FROM entregas AS en
JOIN proveedores AS pv
ON en.id_proveedor=pv.id_proveedor
GROUP BY YEAR(en.fecha_entrega), pv.nombre
ORDER BY YEAR(en.fecha_entrega) ASC;

SELECT * FROM repartos_anuales_por_proveedor;
```

	Año	nombre	Cantidad de Pedidos
▶	2021	AMAZON	1
	2021	UPS	1
	2022	AMAZON	1
	2022	CORREOS	2

CONSULTAS SQL – nº7

7) Cambios en modelo relacional y DB para clasificar a los proveedores de reparto en categorías.

```
-- Creamos la nueva tabla para las categorias de proveedores
CREATE TABLE categoria_prov (
    id_categ_prov VARCHAR(4),
    repartos_anuales VARCHAR(50),
    descripcion VARCHAR (50) AS (CASE
        WHEN id_categ_prov = 'PRO1' THEN 'Proveedor Local'
        WHEN id_categ_prov = 'PRO2' THEN 'Proveedor Regional'
        WHEN id_categ_prov = 'PRO3' THEN 'Proveedor Nacional'
        WHEN id_categ_prov = 'PRO4' THEN 'Proveedor Internacional'
    END),
    PRIMARY KEY (id_categ_prov)
);
```

```
-- Añadimos el campo a la tabla proveedores
ALTER TABLE proveedores
ADD id_categ_prov VARCHAR(4) AFTER id_proveedor;
DESCRIBE proveedores;
```

```
-- Agregamos la foreign key
```

```
ALTER TABLE proveedores
ADD FOREIGN KEY (id_categ_prov) REFERENCES categoria_prov(id_categ_prov);
```

CONSULTAS SQL – nº7

-- Insertamos valores en la nueva tabla

```
INSERT INTO categoria_prov (id_categ_prov, repartos_anuales) VALUES ('PRO1', "Hasta 10.000");
```

```
INSERT INTO categoria_prov (id_categ_prov, repartos_anuales) VALUES ('PRO2', "Hasta 50.000");
```

```
INSERT INTO categoria_prov (id_categ_prov, repartos_anuales) VALUES ('PRO3', "Hasta 100.000");
```

```
INSERT INTO categoria_prov (id_categ_prov, repartos_anuales) VALUES ('PRO4', "Más de 100.000");
```

```
SELECT * FROM categoria_prov;
```

```
UPDATE proveedores SET id_categ_prov = 'PRO4'
```

```
WHERE id_proveedor = 'UE&ESP&004' OR id_proveedor = 'UE&ESP&003' OR id_proveedor = 'UE&ESP&002';
```

```
UPDATE proveedores SET id_categ_prov = 'PRO3'
```

```
WHERE id_proveedor = 'UE&ESP&001';
```

```
SELECT * FROM proveedores;
```

	id_proveedor	id_categ_prov	nombre	calle	numero	telefono	email	ciudad	poblacion
▶	UE&ESP&001	PRO3	CORREOS	AV. DE AMERICA	5	+34612123456	delivery@correos.es	madrid	3300000
	UE&ESP&002	PRO4	UPS	C. DE LOPEZ DE HOYOS	64	+34600011111	delivery@ups.es	madrid	3300000
	UE&ESP&003	PRO4	DHL	C. DEL GENERAL PARDINAS	108	+34612012012	delivery@dhl.es	madrid	3300000
	UE&ESP&004	PRO4	AMAZON	C. SIN NOMBRE	1	+34612333333	delivery@amazon.es	madrid	3300000
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

CONSULTAS SQL – nº8

8) Introducción de nuevos artículos en la tabla artículos

```
ALTER TABLE articulos
```

```
ADD marca_blanca BOOL;
```

```
-- Insertamos un nuevo articulo con la condición de marca blanca como true
```

```
INSERT INTO articulos (id_articulo,descripcion,precio,marca_blanca) VALUES ('004001001','Sudadera gris',15,1);
```

```
SELECT * FROM articulos;
```

	id_articulo	descripcion	precio	marca_blanca
	002001001	Pantalon Masculino Negro	42	NULL
	002001003	Pantalon Masculino Azul	42	NULL
	002002001	Pantalon Femenino Negro	30	NULL
	002002002	Pantalon Femenino Blanco	30	NULL
	003003001	Calcetines unisex negro	8.5	NULL
	003003002	Calcetines unisex blanco	8.5	NULL
	004001001	Sudadera gris	15	1
*	NULL	NULL	NULL	NULL

true

CONSULTAS SQL – nº9

9) Ampliación de la información del proveedor de suministro, incorporando datos de las zonas de cobertura (Países y Regiones)

```
/* Creación de tablas que relacionen proveedores con zonas de cobertura,  
permitiendo que un proveedor tenga varias regiones y países de cobertura*/
```

```
CREATE TABLE zona_cobertura (  
    id_zona_cobertura INT NOT NULL AUTO_INCREMENT,  
    pais_cobertura VARCHAR(50),  
    region_cobertura VARCHAR(50),  
    PRIMARY KEY (id_zona_cobertura)
```

```
);
```

```
CREATE TABLE proveedores_zonas (  
    id INT NOT NULL AUTO_INCREMENT,  
    id_proveedor CHAR(10) NULL,  
    id_zona INT NOT NULL,  
    PRIMARY KEY (id),  
    CONSTRAINT ui_prov_zona UNIQUE (id_proveedor, id_zona)  
);
```

```
ALTER TABLE proveedores_zonas ADD CONSTRAINT fk_prov_zona_prov FOREIGN KEY (id_proveedor) REFERENCES proveedores(id_proveedor);
```

```
ALTER TABLE proveedores_zonas ADD CONSTRAINT fk_prov_zona_zona FOREIGN KEY (id_zona) REFERENCES zona_cobertura(id_zona_cobertura);
```

CONSULTAS SQL – nº9

-- Inserción de países y regiones posibles de cobertura

```
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('España','Europa');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Argentina','America');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Chile','America');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Uruguay','America');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Brasil','America');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Estados Unidos','America');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Alemania','Europa');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Francia','Europa');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Italia','Europa');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Portugal','Europa');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Japon','Asia');
INSERT INTO zona_cobertura (pais_cobertura, region_cobertura) VALUES ('Australia','Oceania');
```

id_zona_cobertura	pais_cobertura	region_cobertura
1	España	Europa
2	Argentina	America
3	Chile	America
4	Uruguay	America
5	Brasil	America
6	Estados Unidos	America
7	Alemania	Europa
8	Francia	Europa
9	Italia	Europa
10	Portugal	Europa
11	Japon	Asia
12	Australia	Oceania
NULL	NULL	NULL

CONSULTAS SQL – nº9

-- Inserción de las zonas de cobertura de cada proveedor

```
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&001',1);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&003',2);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&003',3);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&003',4);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&003',5);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',6);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',1);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',7);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',8);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',9);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&004',10);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&002',11);
INSERT INTO proveedores_zonas (id_proveedor, id_zona) VALUES ('UE&ESP&002',12);
```

```
SELECT prov.nombre AS 'Proveedor', zc.region_cobertura AS 'Region',
       zc.pais_cobertura AS 'Países', cp.descripcion AS 'Categoría'
FROM proveedores_zonas AS pz
JOIN proveedores AS prov
ON pz.id_proveedor=prov.id_proveedor
JOIN zona_cobertura AS zc
ON zc.id_zona_cobertura=pz.id_zona
JOIN categoria_prov AS cp
ON prov.id_catég_prov=cp.id_catég_prov
ORDER BY zc.region_cobertura;
```

Proveedor	Region	Países	Categoría
DHL	América	Argentina	Proveedor Internacional
DHL	América	Chile	Proveedor Internacional
DHL	América	Uruguay	Proveedor Internacional
DHL	América	Brasil	Proveedor Internacional
AMAZON	América	Estados Unidos	Proveedor Internacional
UPS	Asia	Japón	Proveedor Internacional
CORREOS	Europa	España	Proveedor Nacional
AMAZON	Europa	España	Proveedor Internacional
AMAZON	Europa	Alemania	Proveedor Internacional
AMAZON	Europa	Francia	Proveedor Internacional
AMAZON	Europa	Italia	Proveedor Internacional
AMAZON	Europa	Portugal	Proveedor Internacional
UPS	Oceania	Australia	Proveedor Internacional

CONSULTAS SQL – nº10

10) Backup de la tabla pedidos - linea de pedidos diario.

-- Agrego un pedido con fecha de hoy

USE mayorista;

INSERT INTO pedidos (id_pedido, id_sucursal, fecha_pedido) VALUES (9,'00003','2022-10-27 13:59:59');

INSERT INTO linea_pedidos (id_pedido, id_articulo, cantidad) VALUES (9,'001002002',25);

-- Compruebo que se hayan insertado los valores anteriores

SELECT * FROM mayorista.pedidos

WHERE date(fecha_pedido) = curdate();

SELECT ped.id_pedido, lp.id_articulo, lp.cantidad FROM pedidos AS ped

JOIN linea_pedidos AS lp ON ped.id_pedido=lp.id_pedido

WHERE date(ped.fecha_pedido) = curdate();

id_pedido	id_sucursal	fecha_pedido
9	00003	2022-10-27 13:59:59
NULL	NULL	NULL

id_pedido	id_articulo	cantidad
9	001002002	25

CONSULTAS SQL – nº10

```
-- Creo la nueva BBDD que servirá de back-up
CREATE DATABASE BACK_UP;
USE BACK_UP;

-- Creo las tablas mellizas de la tabla pedidos y linea de pedidos de la BBDD mayorista
CREATE TABLE pedidos2 (
    id_pedido VARCHAR(10),
    id_sucursal VARCHAR(6),
    fecha_pedido DATETIME NOT NULL,
    PRIMARY KEY (id_pedido)
);

CREATE TABLE linea_pedidos2 (
    id_pedido VARCHAR(10),
    id_articulo VARCHAR(20),
    cantidad INT,
    FOREIGN KEY (id_pedido) REFERENCES pedidos2(id_pedido)
);
```

CONSULTAS SQL – nº10

```
-- Proceso de Back-Up diario
```

```
DELIMITER $$
```

```
CREATE PROCEDURE respaldo()
```

```
BEGIN
```

```
INSERT INTO back_up.pedidos2
```

```
SELECT * FROM mayorista.pedidos
```

```
WHERE date(mayorista.pedidos.fecha_pedido) = curdate();
```

```
INSERT INTO back_up.linea_pedidos2
```

```
SELECT pd.id_pedido, lp.id_articulo, lp.cantidad FROM mayorista.linea_pedidos as lp
```

```
JOIN mayorista.pedidos as pd
```

```
ON pd.id_pedido=lp.id_pedido
```

```
WHERE date(pd.fecha_pedido) = curdate();
```

```
END $$
```

```
CALL respaldo;
```

```
1 • SELECT * FROM back_up.pedidos2;
```

Result Grid



Filter Rows:

	id_pedido	id_sucursal	fecha_pedido
▶	9	00003	2022-10-27 13:59:59
*	NULL	NULL	NULL

```
1 • SELECT * FROM back_up.linea_pedidos2;
```

Result Grid



Filter Rows:

	id_pedido	id_articulo	cantidad
▶	9	001002002	25

04

CONCLUSIONES FINALES

04. Conclusiones Finales



El proceso de creación de una base de datos implica conocer en detalle la organización y los procesos involucrados en sus operaciones.



Es fundamental tener en claro el input y el output de cada proceso para una eficiente creación del diagrama entidad-relación y modelo relacional.



Una arquitectura de base de datos correctamente propuesta y normalizada nos permite lograr eficiencia en el almacenamiento de datos, tanto a nivel físico como a nivel lógico, y un ahorro de costos y tiempo considerable.

04. Conclusiones Finales



En el caso puntual planteado en la actividad, encontramos la tarea de la creación y adecuación de una BBDD a las operaciones de un mayorista de artículos de vestimenta que comercializa a tiendas en todo el mundo.



Resultó desafiante el poder simular un proceso real de compra con todos sus componentes, integrando las distintas etapas desde la compra hasta su entrega y contemplando distintos escenarios.

GRACIAS

Módulo
Bases de Datos SQL

Prof. Gabriel Marín

Alumno:
Santiago Torres Busquets