

Ejemplo “Temática empresa”

Vemos el modelo relacional del ejemplo empresa (considerando los atributos subrayados como clave primaria y el símbolo (VN) que indica que admite valores nulos):

DEPT (Dept_no, Dnombre, Loc(VN))

EMP (Emp_No, Apellido, Oficio(VN), Jefe(VN), Fecha_Alta(VN), Salario(VN), Comision(VN), Dept_No) donde Dept_No REFERENCIA DEPT y donde Jefe REFERENCIA EMP

CLIENTE (Cliente_Cod, Nombre, Direccion, Ciudad, Estado(VN), Codigo_Postal, Area(VN), Telefono(VN), Repr_Cod(VN), Limite_Credito(VN), Observaciones(VN)) donde Repr_Cod REFERENCIA EMP

PRODUCTO (Prod_Num, Descripcion)

PEDIDO (Ped_Num, Ped_Fecha(VN), Ped_Tipo(VN), Cliente_Cod, Fecha_Pedido(VN), Total(VN)) donde Cliente_Cod REFERENCIA CLIENTE

DETALLE (Ped_Num, Detalle_Num, Prod_Num, Precio_Venta(VN), Cantidad(VN), Importe(VN)) donde Ped_Num REFERENCIA PEDIDO y Prod_Num REFERENCIA PRODUCTO

- Tenemos seis entidades diferentes: departamentos (DEP), empleados (EMP), clientes (CLIENTE), productos (PRODUCTO), órdenes (PEDIDO) y detalle de las órdenes (DETALLE).
- Entre las seis entidades establecen relaciones:
 - Entre DEPT y EMP (relación 1: N), ya que un empleado es asignado obligatoriamente a un departamento, y un departamento tiene asignados cero o varios empleados.
 - Entre EMP y EMP (relación reflexiva 1: N), ya que un empleado puede tener por jefe otro empleado de la empresa, y un empleado puede ser jefe de cero o varios empleados.
 - Entre EMP y CLIENTE (relación 1: N), ya que un empleado puede ser el representante de cero o varios clientes, y un cliente puede tener asignado un representante que debe ser un empleado de la empresa.
 - Entre CLIENTE y PEDIDO (relación 1: N), ya que un cliente puede tener cero o varias órdenes a la empresa, y una orden es obligatoriamente de un cliente.
 - Entre PEDIDO y DETALLE (relación fuerte-débil 1: N), ya que el orden está formada por varias líneas, llamadas detalle de la orden.
 - Entre DETALLE y PRODUCTO (relación N: 1), ya que cada línea de detalle

corresponde a un producto.

- A veces, algún alumno no experto en diseños Entidad-Relación se pregunta el porqué de la entidad DETALLE y piensa que no debería ser, y la sustituye por una relación N: N entre las entidades PEDIDO y PRODUCTO. Gran error. El error radica en el hecho de que en una relación N: N entre PEDIDO y PRODUCTO, un mismo producto no puede estar más de una vez en la misma orden. En ciertos negocios, esto puede ser una decisión acertada, pero no siempre es así, ya que se pueden dar situaciones similares a las siguientes:

- Por razones comerciales o de otra índole, en una misma orden hay cierta cantidad de un producto con un precio y descuentos determinados, y otra cantidad del mismo producto con unas condiciones de venta (Precio y/o descuentos) diferentes.
- Puede que una cantidad de producto deba entregar en una fecha, y otra cantidad del mismo producto en otra fecha. En esta situación, la fecha de envío debería residir en cada línea de detalle.

La implementación de este modelo relacional en MySQL ha provocado las tablas siguientes:

DEPT (Departamento)

Atributo	Null	Tipo	Descripción
DEPT_NO	NOT NULL	INT(2)	Número de departamento
DNOMBRE	NOT NULL	VARCHAR(14)	Descripción
LOC		VARCHAR(14)	Localidad

EMP (Empleado)

Atributo	Null	Tipo	Descripción
EMP_NO	NOT NULL	INT(4)	Número de empleado
APELLIDOS	NOT NULL	VARCHAR(10)	Apellidos
OFICIO		VARCHAR(10)	Oficio
JEFE		INT(4)	Número de empleado que es jefe
FECHA_ALTA		DATE	Fecha de alta
SALARIO		INT(10)	Salario mensual
COMISION		INT(10)	Importe de las comisiones
DEPT_NO	NOT NULL	INT(2)	Departamento al que pertenece

PRODUCTO

Atributo	Null	Tipo	Descripción
PROD_NUM	NOT NULL	INT(6)	Código del producto
DESCRIPCION	NOT NULL	VARCHAR(30)	Descripción del producto

CLIENTE

Atributo	Null	Tipo	Descripción
CLIENTE_COD	NOT NULL	INT(6)	Código de cliente
NOMBRE	NOT NULL	VARCHAR(45)	Nombre del cliente
DIRECCION	NOT NULL	VARCHAR(40)	Dirección del cliente
CIUDAD	NOT NULL	VARCHAR(30)	Ciudad
ESTADO		VARCHAR(2)	País
CODIGO_POSTAL	NOT NULL	VARCHAR(9)	Código Postal
AREA		INT(3)	Área telefónica
TELEFONO		VARCHAR(9)	Teléfono
REPR_COD		INT(4)	Código del empleado que lo representa
LIMITE_CREDITO		DECIMAL(9,2)	Límite de crédito que dispone
OBSERVACIONES		TEXT	Observaciones

PEDIDO (Órdenes de venta)

Atributo	Null	Tipo	Descripción
PED_NUM	NOT NULL	INT(4)	Número pedido
PED_FECHA		DATE	Fecha pedido
PED_TIPO		VARCHAR(1)	Tipo de orden. Valores válidos A, B, C.
CLIENTE_COD	NOT NULL	INT(6)	Código del cliente que hace el pedido
FECHA_ENVIO		DATE	Fecha de envío
TOTAL		DECIMAL(8,2)	Importe total

DETALLE (Detalle de la órdenes de venta)

Atributo	Null	Tipo	Descripción
PED_NUM	NOT NULL	INT(4)	Número de orden de la tabla pedido
DETALLE_NUM	NOT NULL	INT(4)	Número de línea por cada orden
PROD_NUM	NOT NULL	INT(6)	Código del producto de la tabla producto
PRECIO_VENTA		DECIMAL(8,2)	Precio venta del producto
CANTIDAD		INT(8)	Cantidad de producto
IMPORTE		DECIMAL(8,2)	Importe total de la línea

Ejemplo “Temática sanidad”

Vemos el modelo relacional del ejemplo sanidad (considerando los atributos subrayados como clave primaria y el símbolo (VN) que indica que admite valores nulos):

HOSPITAL (Hospital_Cod, Nombre, Direccion(VN), Telefono(VN), Cdad_Camas(VN))
SALA (Hospital_Cod, Sala_Cod, Nombre, Cdad_Camas(VN)) donde Hospital_Cod REFERENCIA HOSPITAL

PLANTILLA (Hospital_Cod, Sala_Cod, Empleado_No, Apellido, Funcion(VN), Turno(VN), Salario(VN)) donde {Hospital_Cod, Sala_Cod} REFERENCIA SALA ENFERMO (Inscripcion, Apellido, Direccion(VN), Fecha_Nac(VN), Sexo, Nss(VN))

INGRESOS (Inscripcion, Hospital_Cod, Sala_Cod, Cama(VN)) donde Inscripcion REFERENCIA ENFERMO, {Hospital_Cod, Sala_Cod} REFERENCIA SALA

DOCTOR (Hospital_Cod, Doctor_No, Apellidos, Especialidad) donde Hospital_Cod REFERENCIA HOSPITAL

- Tenemos seis entidades diferentes: hospitales (HOSPITAL), salas de los hospitales (SALA), doctores de los hospitales (DOCTOR), empleados de las salas de los hospitales (PLANTILLA), enfermos (ENFERMO) y enfermos ingresados actualmente (INGRESOS).

- Entre las seis entidades establecen relaciones:

- Entre HOSPITAL y SALA (relación fuerte-débil 1: N), ya que las salas se identifican con un código de sala dentro de cada hospital; es decir, podemos tener una sala identificada con el código 1 en el hospital X, y una sala identificada también con el código 1 en el hospital Y.

- Entre HOSPITAL y DOCTOR (relación fuerte-débil 1: N), ya que los doctores identifican con un código de doctor dentro de cada hospital; es decir, podemos tener un doctor identificado con el código 10 en el hospital X, y un doctor identificado también con el código 10 en el hospital Y.

- Entre SALA y PLANTILLA (relación fuerte-débil 1: N), ya que los empleados se identifican con un código dentro de cada sala; es decir, podemos tener un empleado identificado con el código 55 en la sala 10 del hospital X, y un empleado identificado también con el código 55 en otra sala de cualquier hospital.

- Entre ENFERMO y INGRESOS (relación 1: 1), ya que un enfermo puede estar ingresado o no.

- Entre SALA y INGRESOS (relación 1: N), ya que en una sala puede haber cero o varios enfermos ingresados, y un enfermo ingresado sólo lo puede estar en una única sala.

- Seguro que no es el mejor diseño para una gestión correcta de hospitales, pero nos interesa mantener este diseño para las posibilidades que nos dará de cara al aprendizaje del lenguaje SQL. Aprovechamos, sin embargo, la ocasión para comentar los puntos oscuros en el diseño:

- Quizás no es muy normal que los empleados de un hospital se identifiquen dentro de cada sala. Es decir, en el diseño, la entidad PLANTILLA es débil de la entidad SALA y, tal vez, sería más lógico que fuera débil de la entidad HOSPITAL de manera similar a la entidad DOCTOR.

- Para poder gestionar los pacientes (ENFERMO) que actualmente están ingresados, es necesario establecer una relación entre ENFERMO y SALA, la

que sería de orden N: 1, ya que en una sala puede haber varios pacientes ingresados y un paciente, si está ingresado, lo está en una sala.

Fijémonos en que la relación (tabla) ENFERMO contiene la pareja de atributos (Hosp_Ingreso, Sala_Ingreso) que conjuntamente son clave foránea de la relación (tabla) SALA y que pueden tener valores nulos (VN), ya que un paciente no debe estar necesariamente ingresado. Si pensamos un poco en la gestión real de estas tablas, nos encontraremos que la tabla ENFERMO normalmente contendrá muchas filas y que, por suerte para los pacientes, muchas de estas tendrán vacíos los campos Hosp_Ingreso y Sala_Ingreso, ya que, del total de pacientes que pasan por un hospital, un conjunto muy pequeño está ingresado en un momento determinado. Esto puede llegar a provocar una pérdida grave de espacio en la base de datos.

En estas situaciones es lícito pensar en una entidad que aglutine los pacientes que están ingresados actualmente (INGRESOS), la cual debe ser débil de la entidad que engloba todos los pacientes (ENFERMO). Esta es la opción adoptada en este diseño.

También sería adecuado disponer de una entidad que aglutinara las diferentes especialidades médicas existentes, de modo que pudiéramos establecer una relación entre esta entidad y la entidad DOCTOR. No es el caso y, por tanto, la especialidad de cada doctor se introduce como un valor alfanumérico.

Asimismo, de manera similar, sería adecuado disponer de una entidad que aglutinara las diferentes funciones que puede llevar a cabo el personal de la plantilla, por lo que pudiéramos establecer una relación entre esta entidad y la entidad PLANTILLA. Tampoco es el caso y, por tanto, la función que cada empleado realiza introduce como un valor alfanumérico.

La implementación de este modelo relacional en MySQL ha provocado las tablas siguientes:

HOSPITAL

Atributo	Null	Tipo	Descripción
HOSPITAL_COD	NOT NULL	INT(2)	Código hospital
NOMBRE	NOT NULL	VARCHAR(10)	Nombre hospital
DIRECCION		VARCHAR(20)	Dirección
TELEFONO		VARCHAR(8)	Teléfono
CDAD_CAMAS		INT(3)	Cantidad de camas

SALA

Atributo	Null	Tipo	Descripción
HOSPITAL_COD	NOT NULL	INT(2)	Código hospital
SALA_COD	NOT NULL	INT(2)	Código sala
NOMBRE	NOT NULL	VARCHAR(20)	Nombre sala
CDAD_CAMAS		INT(3)	Cantidad de camas

DOCTOR

Atributo	Null	Tipo	Descripción
HOSPITAL_COD	NOT NULL	INT(2)	Código hospital
DOCTOR_COD	NOT NULL	INT(3)	Código doctor
APELLIDOS	NOT NULL	VARCHAR(13)	Apellidos
ESPECIALIDAD	NOT NULL	VARCHAR(16)	Especialidad

PLANTILLA

Atributo	Null	Tipo	Descripción
HOSPITAL_COD	NOT NULL	INT(2)	Código hospital
SALA_COD	NOT NULL	INT(2)	Código sala
EMPLEADO_NO		INT(4)	Código empleado
APELLIDOS		VARCHAR(15)	Apellidos
FUNCION		VARCHAR(10)	Función
TURNOS		VARCHAR(1)	Turno
SALARIO		INT(10)	Salario anual

ENFERMO

Atributo	Null	Tipo	Descripción
INSCRIPCION	NOT NULL	INT(5)	Identificación enfermo
APELLIDOS	NOT NULL	VARCHAR(15)	Apellidos
DIRECCION		VARCHAR(20)	Dirección
FECHA_NAC		DATE	Fecha nacimiento
SEXO	NOT NULL	VARCHAR(1)	Sexo: H-hombre, M-mujer
NSS		CHAR(9)	Número Seguridad Social

INGRESOS (enfermos ingresados)

Atributo	Null	Tipo	Descripción
INSCRIPCION	NOT NULL	INT(5)	Identificación enfermo
HOSPITAL_COD	NOT NULL	INT(2)	Código hospital
SALA_COD	NOT NULL	INT(2)	Código sala
CAMA		INT(4)	Número de cama en la sala