**TRABAJO PRÁCTICO**

GESTIÓN DE DATOS

**GRUPO: STR\_NOMBRE\_GRUPO**

**ESTRATEGIA**

|  |  |
| --- | --- |
| INTEGRANTES: |  |
| AGUSTÍN ALABARCE |  |
| ARIEL VILA |  |
| MELANIE GUTERMAN | 143.911-0 |
| OSVALDO FERNÁNDEZ |  |

**Diagrama Entidad – Relación**

Para la resolución del trabajo práctico realizamos un Diagrama Entidad-Relación (DER).

A continuación se detallan las entidades creadas, con sus atributos, cuáles son sus claves primarias (PK), con qué entidades se relacionan mediante claves foráneas (FK), los tipos de datos de cada uno de los atributos y cuando es pertinente realizamos aclaraciones en base a nuestra estrategia de resolución.

**Aclaraciones Generales:**

* Llamamos ID a la clave primaria de cada entidad, como una cuestión de nomenclatura. De acuerdo con cada entidad, la misma puede ser un número autoincremental, un código que se ingresa, o un carácter.
* El campo BAJA es de tipo bit, es decir, que puede estar en 0 o en 1. A todas las entidades que les correspondía este campo, les pusimos que por default el mismo valga 1. Cuando se da de baja lógica a la misma, pasa a valer 1.
* A los campos clave foráneas (FK) decidimos por convención llamarlo por el formato: ENTIDAD\_ID, para darnos cuenta que efectivamente se trata de una clave foránea.

**Entidades**

1. **Usuario**

Atributos:

* Id
* Username
* Password
* Intentos\_login
* Tipo
* Baja

Aclaraciones:

* El campo password es una cadena de caracteres que siempre estará encodeada en SHA-256, es decir, al consultar la tabla en la base de datos no se podrá leer correctamente.
* Intentos\_login empieza en 0 por default. Cuando llega a 3, porque el usuario se loggeó incorrectamente 3 veces, el usuario se inhabilita.
* El campo Tipo es un char, que puede ser ‘C’ (Cliente) o ‘E’ (Empresa) en el TP.

1. **Rol**

Atributos:

* Id
* Nombre
* Baja

Aclaraciones:

* Los roles existentes son “Administrador”, “Cliente” y “Empresa”. No es muy probable que los mismos cambien, pero el enunciado del TP aclara que se podrían crear, modificar o eliminar roles, entonces decidimos tratar al rol como una entidad más.

1. **Usuario\_Rol**

Atributos:

* Usuario\_id
* Rol\_id

Aclaraciones:

* Usuario\_Rol es una entidad muchos-a-muchos. Ambos atributos son claves foráneas de usuario y de rol respectivamente. Decidimos realizarlo de esta manera porque un usuario puede tener más de un rol y viceversa, un rol puede estar asignado a muchos usuarios.

1. **Funcionalidad**

Atributos:

* Id
* Nombre

Aclaraciones:

* Cada objeto de la entidad funcionalidad es una acción que el usuario puede realizar en el sistema.

1. **Rol\_Funcionalidad**

Atributos:

* Rol\_id
* Funcionalidad\_id

Aclaraciones:

* Rol\_Funcionalidad es una entidad muchos-a-muchos. Ambos atributos son claves foráneas de rol y de funcionalidad respectivamente. Decidimos realizarlo de esta manera porque un rol puede tener (y generalmente tiene…) muchas funcionalidades, y una funcionalidad puede ser realizada por más de un rol diferente.

1. **Empresa**

Atributos:

* Id
* Usu\_id
* Razón\_social
* Mail
* Teléfono
* Calle
* Calle\_nro
* Piso
* Dpto
* Localidad
* Cod\_postal
* Ciudad
* Cuit
* Nom\_contacto
* Fecha\_creacion
* Baja

Aclaraciones:

* Una empresa siempre va a estar asociada a un usuario, ya que si un usuario se registra como empresa se crean a la vez ambas entidades, y si el administrador da de alta una nueva empresa, automáticamente se crea la entidad usuario correspondiente a la misma. Por esto, el campo usu\_id (clave foránea correspondiente a una relación uno-a-uno con usuario) es obligatorio.
* Los atributos razón social y CUIT son únicos. Igualmente decidimos crear el campo ID (autoincremental ) como clave primaria, ya que es más costoso relacionar datos string, que ocupan más lugar.
* El campo Depto no es obligatorio.
* Decidimos poner un campo “ciudad” varchar, ya que la tabla maestra no otorgaba datos de la misma, y el TP no incluía búsqueda de empresas/clientes por ciudad, ni se usaba el mismo en ningún lado, en cuyo caso sería más óptimo separar a la ciudad en otra entidad aparte, y relacionar a la empresa mediante una fk con la ciudad correspondiente.

1. **Cliente**

Atributos:

* Id
* Usu\_id
* Nombre
* Ape
* Doc
* Tipo\_Doc
* Mail
* Telefono
* Calle
* Calle\_nro
* Piso
* Depto
* Localidad
* Cod\_Postal
* Fecha\_Nac
* Baja

Aclaraciones:

* Usu\_id: idem empresa.
* El campo “Teléfono” es único.
* La combinación entre los campos Doc y Tipo\_Doc es única.
* El campo Depto no es obligatorio.
* Para resolver el tema de las entidades usuario, cliente y empresa, surgieron varias opciones. La opción elegida nos pareció la más óptima, porque nos daba la posibilidad de relacionar al usuario con la empresa o cliente correspondiente, sin dejar campos null innecesarios, pudiendo acceder a los datos de las tres entidades rápidamente, utilizando únicamente un join.

1. **Publicación**

Atributos:

* Id
* Tipo\_id
* Stock
* Precio
* Estado\_id
* Fecha\_inicio
* Fecha\_finalizacion
* Descripción
* Vis\_id
* Usu\_id
* Permitir\_preguntas

Aclaraciones:

* Publicación es la entidad “central”, que referencia o es referenciada por muchas entidades.
* Decidimos juntar en la misma entidad publicación, a las publicaciones de tipo Subasta y de tipo Compra Inmediata, ya que comparten los atributos.
* El campo Tipo\_id referencia al tipo de publicación, es decir, si se trata de una compra inmediata o de una subasta. El mismo sirve para diferenciar las mismas porque tienen algunas funcionalidades distintas.
* El campo Precio, en el caso de Compra Inmediata referencia al precio por unidad, y en caso de Subasta referencia al precio inicial del lote.
* El campo permitir\_preguntas es un bit, que se encuentra en 1 en caso de que el vendedor decida permitir las preguntas.
* El campo Usu\_id referencia al vendedor que creó la publicación.

1. **Tipo**

Atributos:

* Id
* Descripción

Aclaraciones:

* Esta entidad corresponde a los tipos de publicación posibles, “Compra Inmediata” o “Subasta”.
* Decidimos que el Id sea un char (correspondiente a la letra inicial de la descripción) para que rápidamente y sin tener que joinear, en la publicación se pueda reconocer de qué tipo se trata. La desventaja es que si se agregara algún tipo de publicación que coincida con la inicial de algún tipo existente, habría un inconveniente en la clave primaria. Como el enunciado del TP sólo nos pide que administremos estos dos tipos, y como es muy poco probable que se agregue otro, seguimos esta estrategia.

1. **Estado**

Atributos:

* Id
* Descripción

Aclaraciones

* Entidad que corresponde al estado posible de una publicación, “Borrador”, “Activa”, “Pausada” y “Finalizada”.
* Al igual que en el tipo, decidimos que el campo Id sea un char.

1. **Pregunta**

Atributos:

* Id
* Pub\_id
* Usu\_id
* Texto

Aclaraciones:

* Una publicación puede tener más de una pregunta, pero una pregunta pertenece a una sóla publicación (relación muchos a uno). Es por esto que referenciamos a la publicación desde la entidad Pregunta.
* El campo usu\_id corresponde al usuario que realizó la pregunta.

1. **Respuesta**

Atributos:

* Id
* Texto
* Fecha
* Pre\_id

Aclaraciones:

* La relación pregunta – respuesta es uno-a-uno, ya que una pregunta sólo puede tener una respuesta. Separamos las entidades ya que si poníamos el texto y la fecha de respuesta como atributos de Pregunta, éstos iban a quedar en null hasta que la misma sea respondida, lo que no nos parecía óptimo.
* No nos pareció necesario agregar en esta entidad el id del que usuario que respondió, porque siempre va a ser el mismo que creó la publicación, entonces se puede obtener dicha información muy fácil.

1. **Oferta**

Atributos:

* Id
* Pub\_id
* Monto
* Fecha
* Usu\_id
* Cal\_id

Aclaraciones:

* Esta entidad se corresponde con las publicaciones de tipo Subasta.
* El campo usu\_id hace referencia al usuario que realizó la oferta.
* En el sistema se comprueba que el monto de la oferta nueva sea mayor al de la última oferta realizada para esa publicación. También se debe comprobar que, de no existir ofertas anteriores, el monto sea mayor que el precio inicial del lote.
* El campo cal\_id corresponde a la entidad Calificación. En caso de que no se haya calificado al vendedor todavía, o que la oferta no sea la ganadora de la subasta, estaría en null.

1. **Compra**

Atributos:

* Id
* Pub\_id
* Cantidad
* Fecha
* Usu\_id
* Cal\_id

Aclaraciones:

* Parecido a la entidad Oferta, pero se utiliza para las publicaciones de tipo “Compra Inmediata”.
* El atributo Cantidad corresponde a la cantidad comprada, que debe ser menor al stock disponible en el momento de la compra.
* No es necesario el atributo Monto, ya que se calcula como la cantidad \* el precio por unidad.

1. **Calificación**

Atributos:

* Id
* Cant\_estrellas
* Descripción

Aclaraciones:

* Entidad referenciada desde Oferta (ganadora) o Compra. La misma no contiene la clave foránea del vendedor que está siendo calificado, pero se puede obtener joineando con la entidad Publicación.

1. **Rubro**

Atributos:

* Id
* Descripción
* Baja

Aclaraciones:

* Entidad que corresponde a los rubros a los que puede estar asociada una publicación, como por ejemplo: “Industrias y Oficinas”, “Música, Películas y Series”, y “Electrónica, Audio y Video”.

1. **Publicacion\_Rubro**

Atributos:

* Rubro\_id
* Pub\_id

Aclaraciones:

* Dado que una publicación puede tener muchos rubros, y un rubro (lógicamente) puede estar asignado a muchas publicaciones, se creó esta entidad que corresponde a la relación muchos-a-muchos existente entre las entidades Publicación y Rubro.

1. Item\_Factura

Atributos:

* Id
* Monto
* Cantidad
* Pub\_id
* Fac\_id

Aclaraciones:

* El ítem de una factura está asociado directamente a una publicación, pero puede haber varios objetos Item\_Factura por publicación: por ejemplo, si cobro la comisión por una visibilidad, y además dos compras, la primera de tres productos y la segunda de dos, tendría tres Items\_Factura diferentes. Por esto le agregamos un Id autoincremental como primary key.

1. **Factura**

Atributos:

* Id
* Fecha
* Total
* Usu\_id
* Forma\_id

Aclaraciones:

* El campo Total está desnormalizado. Siempre va a tener que coincidir con la suma de los montos de los ítem\_factura pertenecientes a la factura. Nos pareció conveniente dejarlo desnormalizado ya que se necesita consultar, por ejemplo, los clientes que tuvieron mayor facturación, para lo cual se necesita calcular el total de facturación de cada cliente. Dicho procedimiento es mucho más costoso si tenemos que recorrer todos los ítems de la factura.
* Por el mismo motivo, decidimos crear el campo usu\_id, aunque se puede llegar a el (usuario al que le facturamos) a través de cualquier ítem de la factura, relacionándolo con la publicación. Esto sería mucho más costoso que si repetimos el campo usu\_id en la entidad Factura.
* Forma\_Id referencia a la forma de pago elegida por el usuario.