

Fundamentos de Bases de Datos

Practica 7

Normalización de la Base de Datos

16 de Abril del 2019

El objetivo en esta práctica es normalizar la base de datos de acuerdo al caso de uso que hemmos venido usando. Normalizar la base de datos evita almacenar información redundante, para esta practica se usa la tercera formal normal (3NF).

1. Esquema relacional no normalizado

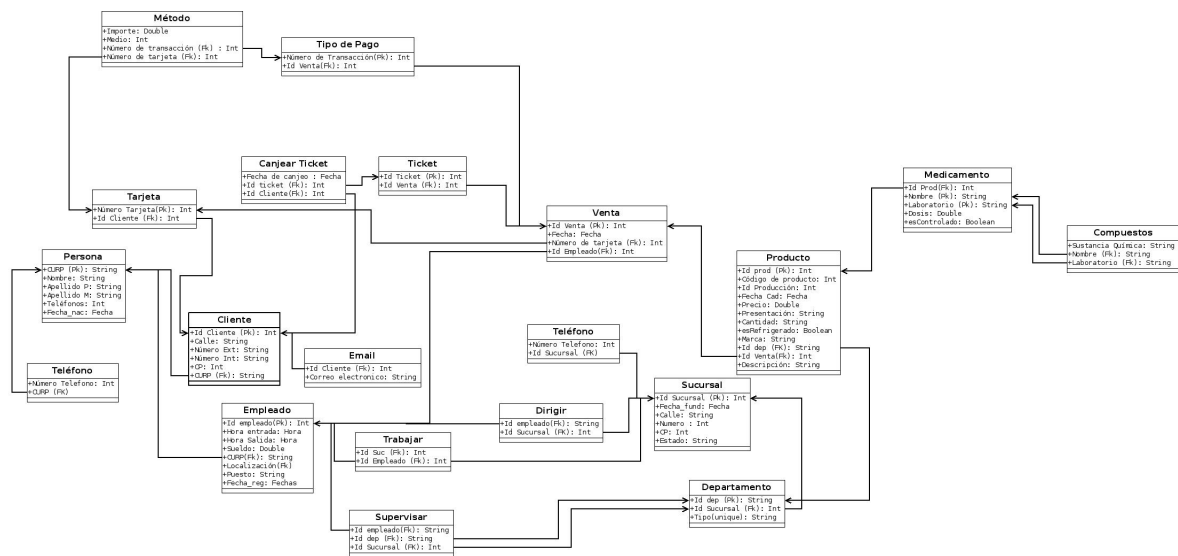


Figura 1: Esquema de la práctica anterior

1.1. Cambios antes de la normalización

- Se trasforman la relación de Departamento a una relación de pertenencia entre una sucursal y un catálogo de tipos.
- Se corrigieron algunos errores de coincidencia de tipo, coincidencia de nombre.

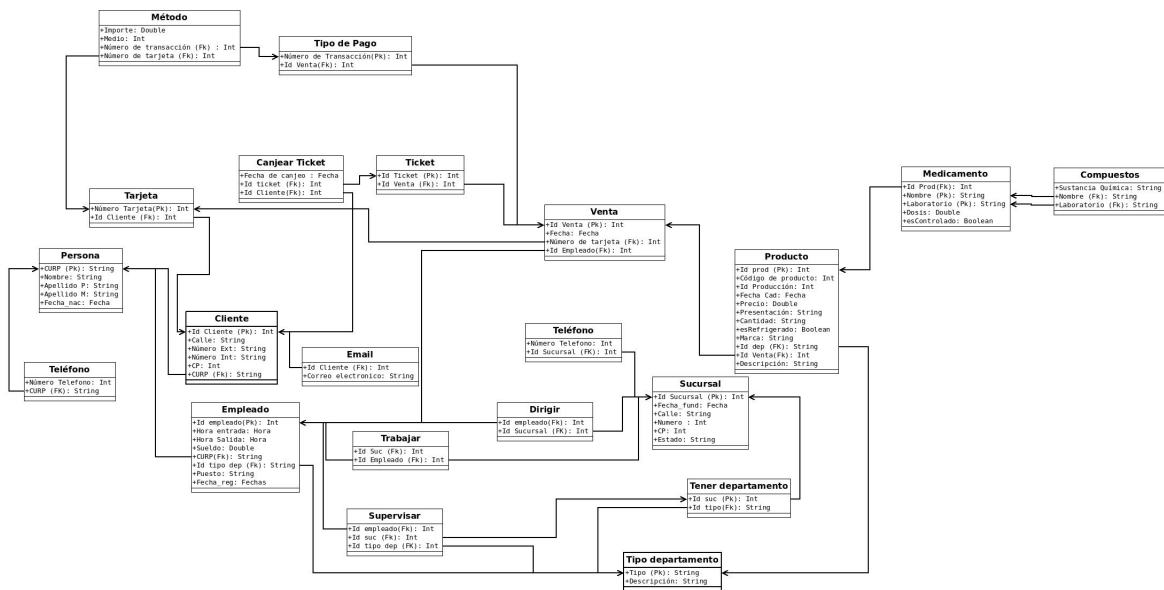


Figura 2: Esquema justo antes de la normalización

2. Normalizando relaciones

2.1. Tablas claramente ya normalizadas

Todas las relaciones con únicamente dos atributos ya está normalizadas, pues las únicas dependencias funcionales que puede haber son las triviales o las inducidas por llaves (un atributo determina al otro), y ninguna de estas es violación a la tercera forma normal. En el esquema, estas son

- Tipo de pago
- Tarjeta
- Ticket
- Teléfono sucursal
- Teléfono persona
- Email
- Trabajar
- Dirigir
- Tener departamento
- Tipo departamento

2.2. Tablas no tan claramente ya normalizadas

1. Método

Tanto el importe como el medio sólo son valores para representar la cantidad de dinero, que no son únicos entre transacciones.

Una transacción puede tener varios métodos de pago, así que tampoco puede determinar nada extra.

Y una tarjeta puede tener varias instancias de método de pago, por lo que tampoco puede identificar a nada extra.

Entonces las únicas dependencias funcionales que hay son triviales, por lo que la relación ya está normalizada.

2. Canjear Ticket

Pueden haber muchos canjeos el mismo día, así que la fecha no determina nada extra.

Un cliente canjea varios tickets, así que tampoco el cliente determina nada extra. Y se pueden canjear varios tickets el mismo día, así que la fecha con el cliente tampoco determina nada extra.

El identificador del ticket es único, pues un ticket solo se canjea una vez. Entonces el identificador del ticket determina a todo lo demás, por lo que es llave.

Entonces, si la relación es

$$\overbrace{\text{CanjearTicket}}^{\text{Ct}}(\overbrace{\text{Fecha}}^F, \overbrace{\text{Id_Ticket}}^T, \overbrace{\text{Id_Cliente}}^C) = \text{Ct}(F, T, C)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{T \rightarrow CF\}$$

Que no violan la tercera forma normal (todas tiene una llave a la izquierda) .

Por lo que la relación ya está normalizada.

3. Venta

Puede haber varias ventas por día, así que la fecha no determina nada extra.

Una tarjeta puede usarse para varias ventas, así que el número de tarjeta no determina nada extra. Esas ventas pueden ser el mismo día, por lo que la fecha junto con el número de tarjeta no determina nada extra.

Un empleado puede atender varias ventas así que el identificador de cliente no determina nada extra. También puede atender varias ventas el mismo día, y ventas diferentes que usaron la misma tarjeta así que el empleado junto con la fecha o la tarjeta no determinan nada extra.

El identificador de venta es la llave, así que determina todo.

Entonces, si la relación es

$$\overbrace{\text{Venta}}^{\text{V}}(\overbrace{\text{Id_Venta}}^I, \overbrace{\text{Fecha}}^F, \overbrace{\text{Num_Tarjeta}}^T, \overbrace{\text{Id_Empleado}}^E) = \text{V}(I, F, T, E)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{I \rightarrow FTE\}$$

Y como todas tienen una llave a la izquierda, no son violación de la tercera forma normal.

Entonces la relación ya está normalizada.

4. Medicamento

El laboratorio puede tener varios medicamentos, así que no puede determinar nada extra.

La dosis tampoco puede determinar nada extra, pues puede repetirse, al igual que la dosis con el laboratorio.

Igualmente, la bandera sobre si es controlado no determina nada extra, así como tampoco esa bandera junto con el nombre del laboratorio o la dosis, o las tres juntas.

El identificador del producto es único por lo que determinan todo y es llave , al igual que el nombre del medicamento.

Entonces, si la relación es

$$\overbrace{\text{Medicamento}}^{\text{M}}(\overbrace{\text{Id_Prod}}^I, \overbrace{\text{Nombre}}^N, \overbrace{\text{Laboratorio}}^L, \overbrace{\text{Dosis}}^D, \overbrace{\text{Es_Controlado}}^C) = \text{M}(I, N, L, D, C)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{I \rightarrow NLDC, N \rightarrow ILDC\}$$

Y como todas tienen una llave a la izquierda, no son violación de la tercera forma normal.

Entonces la relación ya está normalizada.

5. Compuestos

El nombre y el laboratorio son parte de una llave foránea de medicamento. Un medicamento puede tener varios compuesto, así que no determina nada extra.

La sustancia puede estar en varios medicamentos, así que tampoco es única, por lo que no puede determinar nada extra.

Entonces sólo hay dependencias funcionales triviales y la relación ya está normalizada.

6. Supervisar

Un empleado puede supervisar varios departamentos, por lo que no determina nada extra.

Un mismo tipo de departamento puede estar en varias sucursales, y un mismo empleado puede supervisar el mismo departamento en sucursales distintas, por lo que el departamento no determina nada, así como tampoco lo hacen el departamento y el empleado.

La sucursal puede tener varios departamentos, así que no puede determinar nada extra. Y en una misma sucursal un empleado puede supervisar varios departamentos, por lo que la sucursal y el empleado tampoco determinan nada extra.

Cada departamento en una sucursal dada tiene sólo un supervisor, por lo que el tipo de departamento con la sucursal determinan el empleado supervisor. Y como esta dependencia incluye a todos los atributos de la relación, es llave.

Entonces, si la relación es

$$\overbrace{\text{Supervisar}}^S(\overbrace{Id_Empleado}^E, \overbrace{Id_Departamento}^D, \overbrace{Id_Sucursal}^{Su}) = S(E, D, Su)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{DSu \rightarrow E\}$$

Y como la parte izquierda es una llave, no se viola la tercera forma normal.

Por lo que la relación ya está normalizada.

7. Persona

Tanto el Nombre como los dos Apellidos sólo son la representación atómica del atributo de Nombre Completo, que puede no ser único, así que no determina nada extra.

De igual manera, varias personas nacen el mismo día, así que esta tampoco determina nada extra.

Y sería demasiada coincidencia, pero es posible que personas con el mismo nombre nascan el mismo día, así que el nombre completo con la fecha de nacimiento no determina nada extra.

La CURP es la llave, por lo que determina todo lo demás.

Entonces, si la relación es

$$\overbrace{\text{Persona}}^P(\overbrace{CURP}^C, \overbrace{Nombre}^N, \overbrace{ApellidoP}^{Pa}, \overbrace{ApellidoM}^M, \overbrace{Fecha_Nac}^F) = P(C, N, Pa, M, F)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{C \rightarrow NPaMF\}$$

Y como la parte izquierda es llave, no hay violaciones a la tercera forma normal.

Entonces la relación ya está normalizada.

8. Cliente

Tanto la calle, como los números y el CP son representación atómica del atributo de dirección. Y como puede haber varios clientes en una misma dirección, esta no determina nada extra.

Tanto la CURP como el identificador de cliente son únicos, así que son llave.

Entonces, si la relación es

$$\overbrace{\text{Cliente}}^C(\overbrace{Id_Cliente}^I, \overbrace{Calle}^{Ca}, \overbrace{NumeroExterno}^{Ne}, \overbrace{NumeroInterno}^{Ni}, \overbrace{CP}^{Cp}, \overbrace{CURP}^{Cu}) = C(I, Ca, Ne, Ni, Cp, Cu)$$

Las dependencias funcionales no triviales son

$$\mathcal{F} = \{I \rightarrow CaNeNiCpCu, Cu \rightarrow ICaNeNiCp\}$$

Y como la parte izquierda de ambas es llave, no hay violaciones a la tercera forma normal. Entonces la relación ya está normalizada.

2.3. Relacion Producto

Para facilitar la definición de las dependencias funcionales de la relación Producto nos conviene etiquetarla de la siguiente manera:

$$\begin{aligned} & \overbrace{\text{PRODUCTO}}^{\mathbf{P}} (\overbrace{Id_Producto}^{\mathbf{A}}, \overbrace{Codigo_Barras}^{\mathbf{B}}, \overbrace{Id_Produccion}^{\mathbf{C}}, \overbrace{Fecha_Cad}^{\mathbf{D}}, \overbrace{Precio}^{\mathbf{E}}, \overbrace{Presentacion}^{\mathbf{F}}, \\ & \quad \overbrace{Cantidad}^{\mathbf{G}}, \overbrace{esRefrigerado}^{\mathbf{H}}, \overbrace{Marca}^{\mathbf{I}}, \overbrace{Id_Departamento}^{\mathbf{J}}, \overbrace{Id_Venta}^{\mathbf{K}}, \overbrace{Descripcion}^{\mathbf{L}}) \\ & = \mathbf{P}(\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}, \mathbf{E}, \mathbf{F}, \mathbf{G}, \mathbf{H}, \mathbf{I}, \mathbf{J}, \mathbf{K}, \mathbf{L}) \end{aligned}$$

Las siguientes dependencias funcionales para esta relacion son:

$$\mathcal{F} = \{\text{BIFG} \rightarrow \text{E}, \text{BC} \rightarrow \text{D}, \text{B} \rightarrow \text{FGHI}\}$$

Para normalizar primero tenemos que calcular la cerradura para cada dependencia funcional en \mathcal{F} ,

$$\{\text{BIFG}\}^+ = \{\text{BIFGEH}\}$$

$$\{\text{BC}\}^+ = \{\text{BCFGHIDE}\}$$

$$\{\text{B}\}^+ = \{\text{BFGHIE}\}$$

Observamos que a la cerradura de BC es que contiene el mayor numero de atributos y unicamente le faltan algunos atributos, por lo tanto una llave para \mathbf{P} es ABCJKL

Lo siguiente es buscar violaciones a la tercera forma normal, para eso hay que ver que no aparezca la llave en las dependencias funcionales, observamos que las tres dependencias son violaciones a la 3NF, así que se toma una dependencia funcional con mas de un atributo a la izquierda para ver si alguno de los atributos es superfluo.

2.3.1. Superfluos a la Izquierda

Tomamos $\text{BIFG} \rightarrow \text{E}$, luego verificamos si algun atributo a la izquierda es superfluo.

- ¿B es superfluo? $\text{IFG} \rightarrow \text{E}$

$$\{\text{IFG}\}^+ = \{\text{IFG}\}$$

Como E no aparece en la cerradura de IFG, por lo tanto se concluye que B no es superfluo.

- ¿I es superfluo? $\text{BFG} \rightarrow \text{E}$

$$\{\text{BFG}\}^+ = \{\text{BFGHIE}\}$$

E aparece en la cerradura de BFG, por lo tanto se concluye que I es superfluo.

- ¿F es superfluo? $BIG \rightarrow E$

$$\{BIG\}^+ = \{BIGFHE\}$$

E aparece en la cerradura de BIG, por lo tanto se concluye que F es superfluo.

- ¿G es superfluo? $BIF \rightarrow E$

$$\{BIF\}^+ = \{BIFGHE\}$$

E aparece en la cerradura de BIF, por lo tanto se concluye que G es superfluo.

Como resultado de buscar superfluos por la izquierda obtenemos una \mathcal{F} nueva,

$$\mathcal{F} = \{B \rightarrow E, BC \rightarrow D, B \rightarrow FGHI\}$$

y por la regla de la union obtenemos

$$\mathcal{F} = \{BC \rightarrow D, B \rightarrow FGHI\}$$

2.3.2. Superfluos por la derecha

Tenemos a $B \rightarrow FGHI$ y buscamos elementos superfluos a la derecha.

- ¿F es superfluo? $B \rightarrow GHIE$

$$\mathcal{F}' = \{BC \rightarrow D, B \rightarrow GHIE\}$$

Calculamos la cerradura de B usando a \mathcal{F}'

$$\{B\}^+ = \{BGHIE\}$$

F no aparece en la cerradura de B, por lo tanto F no es superfluo.

- ¿G es superfluo? $B \rightarrow FHIE$

$$\mathcal{F}' = \{BC \rightarrow D, B \rightarrow FHIE\}$$

Calculamos la cerradura de B usando a \mathcal{F}'

$$\{B\}^+ = \{BFHIE\}$$

G no aparece en la cerradura de B, por lo tanto G no es superfluo.

- ¿H es superfluo? $B \rightarrow FGIE$

$$\mathcal{F}' = \{BC \rightarrow D, B \rightarrow FGIE\}$$

Calculamos la cerradura de B usando a \mathcal{F}'

$$\{B\}^+ = \{BFGIE\}$$

H no aparece en la cerradura de B, por lo tanto H no es superfluo.

- ¿I es superfluo? $B \rightarrow FGHE$

$$\mathcal{F}' = \{BC \rightarrow D, B \rightarrow FGHE\}$$

Calculamos la cerradura de B usando a \mathcal{F}'

$$\{B\}^+ = \{BFGHE\}$$

I no aparece en la cerradura de B, por lo tanto I no es superfluo.

- ¿E es superfluo? $B \rightarrow FGHI$

$$\mathcal{F}' = \{BC \rightarrow D, B \rightarrow FGHI\}$$

Calculamos la cerradura de B usando a \mathcal{F}'

$$\{B\}^+ = \{BFGHI\}$$

E no aparece en la cerradura de B, por lo tanto E no es superfluo.

Así que la \mathcal{F}_{min} es,

$$\mathcal{F}'_{min} = \{BC \rightarrow D, B \rightarrow FGHE\}$$

Esto quiere decir que las relaciones quedan como:

$R_1(B, C, D)$ con $BC \rightarrow D$

$R_2(B, F, G, H, I, E)$ con $B \rightarrow FGHE$

El siguiente paso es verificar si la llave esta en alguna relaciones obtenidas, como la llave no aparece en ninguna relación la solución es agregar otra relación que la contenga.

$R_3(A, B, C, J, K, L)$ con $ABCJKL \rightarrow ABCJKL$

∴ R_1, R_2, R_3 es la normalización en 3NF para la relación Producto.

2.4. Relacion Empleado

Para la relación empleado se tiene:

$$\begin{aligned} & \overbrace{\text{EMPLEADO}}^E (\overbrace{Id_empleado}^{ie}, \overbrace{Hora_entrada}^{he}, \overbrace{Hora_salida}^{hs}, \overbrace{Sueldo}^S, \overbrace{CURP}^C, \overbrace{Id_tipo_dep}^{td}, \overbrace{Puesto}^P, \overbrace{Fecha_reg}^{fr}) \\ &= \mathbf{E}(ie, he, hs, S, C, td, P, fr) \end{aligned}$$

Se definen las siguientes dependencias funcionales para la relación Empleado

$$\mathcal{F} = \{tdfrP \rightarrow S, tdP \rightarrow hehs\}$$

Calculamos la cerradura para cada una de las dependencias en \mathcal{F}

$$\{tdfrP\}^+ = \{tdfrPShes\}$$

$$\{tdP\}^+ = \{tdPhehs\}$$

Una llave para **E** es ieCtdfrP.

Buscamos violaciones a la tercera forma normal, y encontramos que las dos son violaciones.

2.4.1. Superfluos a la Izquierda

Para buscar elementos superfluos a la izquierda elegimos la siguiente dependencia : $tdfrP \rightarrow S$

- ¿td es superfluo? $frP \rightarrow S$
 $\{frP\}^+ = \{frP\}$

S no aparece en la cerradura de frP , por lo tanto td NO es superfluo.

- ¿fr es superfluo? $tdP \rightarrow S$
 $\{tdP\}^+ = \{tdPhehs\}$

S no aparece en la cerradura de tdP , por lo tanto fr NO es superfluo.

- ¿P es superfluo? $tdfr \rightarrow S$
 $\{tdfr\}^+ = \{tdfr\}$

S no aparece en la cerradura de $tdfr$, por lo tanto P NO es superfluo.

$$\mathcal{F}_{min} = \{tdfrP \rightarrow S, tdP \rightarrow hehs\}$$

2.4.2. Superfluos a la Derecha

Para buscar atributos superfluos por la derecha elegimos la siguiente dependencia que tiene mas de un atributo a la derecha : $tdP \rightarrow hehs$

- ¿he es superfluo? $tdP \rightarrow hs$

Se obtiene una $\mathcal{F}' = \{tdfrP \rightarrow S, tdP \rightarrow hs\}$ y de aquí calculamos la cerradura para tdP
 $\{tdP\}^+ = \{tdPhs\}$ he no aparece en la cerradura de tdP , por lo tanto he NO es superfluo.

- ¿hs es superfluo? $tdP \rightarrow he$

De aquí definimos una $\mathcal{F}' = \{tdfrP \rightarrow S, tdP \rightarrow he\}$ y calculamos la cerradura para tdP
 $\{tdP\}^+ = \{tdPhe\}$ hs no aparece en la cerradura de tdP , por lo tanto hs NO es superfluo.

Por lo tanto se concluye que la \mathcal{F}_{min} es:

$$\mathcal{F}_{min} = \{tdfrP \rightarrow S, tdP \rightarrow hehs\}$$

así que las relaciones serían

$$\begin{aligned} R_1(td, fr, P, S) & \text{ con } tdfrP \rightarrow S \\ R_2(td, P, he, hs) & \text{ con } tdP \rightarrow hehs \end{aligned}$$

Pero de estas dos relaciones ninguna de ellas contiene a la llave, entonces agregamos una nueva relación que la contenga $R_3(ie, C, td, fr, P)$ con $ieCtdfrP \rightarrow ieCtdfrP$.

$\therefore R_1, R_2, R_3$ es la normalización en 3NF para la relación Empleado.

2.5. Relacion Sucursal

La relación es

$$\overbrace{\text{Sucursal}}^S(\overbrace{Id_Suc}^I, \overbrace{Fecha_Fund}^F, \overbrace{Calle}^C, \overbrace{Numero}^N, Cp, \overbrace{Estado}^E) = \mathbf{S}(I, F, C, N, Cp, E)$$

Con las siguientes dependencias funcionales

$$\mathcal{F} = \{I \rightarrow FCNCpE, Cp \rightarrow E\}$$

Si bien existen dependencias funcionales que violan la tercera forma normal, sólo es una dependencia funcional que sólo concierne a dos atributos, por lo que se consideró que no vale la pena normalizar esta relación, puesto que el trabajo necesario para hacerlo es demasiado en comparación a la disminución de redundancia obtenida.

2.6. Esquema normalizado

En este esquema realizamos un par de cambios posteriores a la normalización que surgieron cuando estábamos trabajando con el script de DDL debido a que algunas llaves foráneas eran datos a los que no podíamos aplicar un CONSTRAINT adecuado. Estos cambios fueron:

- Horario tiene a Id.tipo_dep como (Fk) y como (Pk) compuesta a (Id.tipo_dep, puesto).
- Sueldo tiene como (Fk) proveniente de Departamento a Id.tipo_dep , apunta a Horario usando (Fk) (tipo_dep,puesto) y su (Pk) es (Id.tipo_dep , puesto, fecha_registro).
- Empleado tiene como (Fk)'s los atributos (Id.tipo_dep,puesto,fecha_registro).

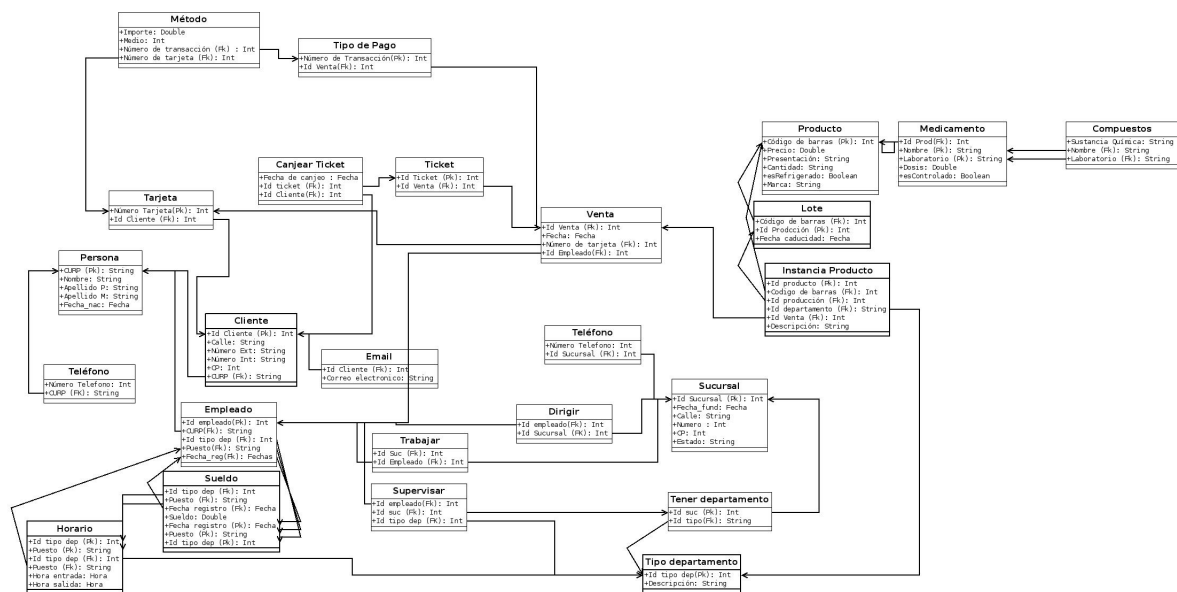


Figura 3: Esquema normalizado

Preguntas extra

Oracle Database Types

Para responder a esto, consultamos en la página de Oracle la documentación de las versiones usadas, específicamente bajo las guías de *Administración* en el apartado de **Database Concepts**:

- [Oracle DB Enterprise Edition 18c](#).
- [Oracle DB Enterprise Edition 11g, Release 1](#).

En la versión 18c

Se listan las siguientes categorías de tipos de datos:

- Character Data Types [CHAR, VARCHAR2, NCHAR, NVARCHAR2]
- Numeric Data Types [NUMBER, BINARY_FLOAT, BINARY_DOUBLE]
- Datetime Data Types [DATE, TIMESTAMP, DATETIME(2)]
- Rowid Data Types [ROWID(categorizados como físicos, lógicos, foráneos ó universales)]
- Format Models and Data Types `// 'moldes'` de los tipos de datos como cadenas, metadatos

Aparte de estos, también menciona la existencia de *raw*, *large objects (LOBs)*, y *collections*, junto con los tipos que se pueden acceder a través de PL/SQL. ¹

En la versión 11g

Se pueden encontrar las siguientes categorías y tipos de datos:

- Character Data Types [CHAR, VARCHAR2, NCHAR, NVARCHAR2, LONG]
- Numeric Data Types [NUMBER, BINARY_FLOAT, BINARY_DOUBLE]
- Datetime Data Types [DATE, TIMESTAMP, DATETIME(2)]

¹variables, 'constantes' (Booleanos?), referencias y tipos, tanto compuestos como definidos por el usuario.

- LOBs [BLOB, CLOB, NCLOB, BFILE]
- RAWs [RAW, LONG_RAW]
- Rowid Data Types [ROWID, UROWID]

Vale la pena notar que LOB y RAW también existen en la versión 18c, aunque sólo se mencione su existencia. En esta documentación también aparecen como categorías los *ANSI*, *DB2*, *SQL/DS*, *XML*, *URI*, *Object* y *Object Views* pero recuperando la idea de la documentación de la versión 18c, decidimos no listarlos como a los anteriores.

Tamaño de nombres internos

Para responder a esto, consultamos en la página de Oracle la documentación de las versiones usadas, específicamente bajo las guías de *Database Reference* fijándonos en los *static data dictionary*:

- [Oracle DB Enterprise Edition 18c](#) ².
- [Oracle DB Enterprise Edition 10g, Release 2](#).

En ambas secciones está especificado que los nombres de columna son VARCHAR2, i.e. cadenas de longitud variable de a lo más cierto número de caracteres. Para la versión 10 (en general, lo mismo que las versiones anteriores a la 12) el límite es de 30, mientras que en la versión 18c es de 128 (en general, lo mismo que las versiones posteriores a la 12).³

CONSTRAINT's en SQL

Esta pregunta era referente a '¿Cuántos tipos de CONSTRAINT hay en SQL?', de acuerdo con [el capítulo 8](#) hay 6:

- NOT NULL exige que no haya null entre los valores de la columna.
- UNIQUE exige que no haya repeticiones entre los valores de la columna.
- PRIMARY KEY combina las exigencias de las dos anteriores.
- FOREIGN KEY pide que los valores de la columna coincidan con al menos uno en una columna de otra tabla.
- CHECK exige que cada valor de la columna cumpla una condición booleana expresado por medio de una cláusula CHECK.
- REF la versión corta según lo que leí es que es un tipo de condición que se puede usar para especificar propiedades que deben cumplir los valores relacionados entre una columna y otra a la que hace un REF.

²En el capítulo 3: *Static Data Dictionary Views: ALL_PART.COL_STATISTICS to DATABASE_PROPERTIES*

³Estos tamaños son los mismos en los límites de otras cosas de la BD como nombres de atributos, es decir, los tamaños no son específicos a columnas. Estos datos se pueden checar en los mismos documentos vinculados pero en las tablas adecuadas. Ej. *ALL_ATTRIBUTE_DIM_ATTR_CLASS*