ÁLVARO RODRÍGUEZ MOLINA

PROYECTO RECUPERACIÓN DE LA ASIGNATURA BASE DE DATOS

1ºdesarrollo de aplicaciones web – tarde

IES ALIXAR – Castilleja de la Cuesta

- La Aplicación consiste en poder acceder a una base de datos con MySQL para realizar consultas y dar información de los clientes de una Asesoría que trabaja con todas las compañías eléctricas

ÍNDICE

(proyecto final)

NOTA: el logo del centro arriba a la izquierda retorna a la página principal

PROYECTO BASE DE DATOS
Álvaro Rodríguez Molina

LinkedIn GitHub

TITULO Y DESCRIPCIÓN DEL PROYECTO:

TÍTULO: StrattonApp

DESCRIPCIÓN: La Aplicación consiste en poder acceder a una base de datos con MySQL para realizar consultas y dar información de los clientes de una compañía Eléctrica

Herramientas:

Diagrama MER extendido - Diagrams

Generar datos aleatorios – Kaggle

Diseño – Canva

Animar diseños – Figma

Animar contenido - Jitter

Imagenes de stok – Unsplash

Video de stock - Pixabay

Colores en tendencia – Coolors

Fuentes que combinan – Fontjoy

Iconos gratis – Icons8

Inspiración creativa – Dribbble

FASE 1: RECOGIDA Y ANÁLISIS DE REQUISITOS

Recolección de requisitos funcionales:

RF-1: el sistema debe permitir registrar a los clientes recogiendo sus datos

RF-2: el sistema debe conocer a que comercializadora pertenece dicho cliente y cual es el nombre de su oferta (idPlan)

RF-3: la información de los clientes partirá de sus números de teléfono, siendo estos únicos para la identificación de la información

RF-4: las ubicaciones completas como Comunidad autónoma, provincia y municipio, se verán determinadas por el codigo postal (ZIP)

RF-5: cada cliente tendrá de manera obligatoria uno o más CUPS_luz como puntos de suministros en cada cual con su UBICACION.

RF-6: El sistema debe permitir que los empleados estén asociados a una asesoría.

RF-7: El sistema debe permitir que los clientes tengan información bancaria asociada.

RF-8: El sistema debe permitir que los clientes tengan múltiples CUPS_gas como puntos de suministro.

RF-9: El sistema debe permitir que los empleados estén asociados a un cliente en particular.

RF-10: El sistema debe permitir la gestión de contratos de clientes, incluyendo la fecha de inicio y finalización del contrato, así como el estado del mismo (activo, inactivo, cancelado, etc.).

Formalización de los requisitos:

RF-1: Registro de clientes

Descripción: El sistema debe permitir el registro de clientes recopilando su información personal.

Atributos asociados: idCliente (identificador único), telefono (único) correo (único), nombre, DNI (unico), dirección, fecha_nacimiento.

RF-2: Asociación de clientes con comercializadora y plan

Descripción: El sistema debe registrar la asociación de cada cliente con una comercializadora y el plan que ha seleccionado.

Atributos asociados: idCliente (referencia al cliente), idComercializadora (referencia a la comercializadora), idPlan (referencia al plan)

RF-3: Identificación de clientes por número de teléfono

Descripción: El sistema debe utilizar el número de teléfono como identificador único para la información de los clientes.

Atributos asociados: telefono (único), …

RF-4: Determinación de ubicaciones a partir del código postal

Descripción: El sistema debe determinar la ubicación completa, incluyendo comunidad autónoma, provincia y municipio, a partir del código postal.

Atributos asociados: ZIP (identificador único), CA, provincia, municipio.

RF-5: Gestión de puntos de suministro de energía

Descripción: Cada cliente debe tener uno o más puntos de suministro de energía (CUPS_luz) asociados, junto con su ubicación.

Atributos asociados: idCliente (referencia al cliente), CUPS_luz (único), ZIP (referencia a la ubicación),

RF-6: Asignación de empleados a asesorías

Descripción: Los empleados deben estar asociados a una asesoría específica.

Atributos asociados: idAgent (identificador único del empleado), codigoempleado, idAsesoria (referencia a la asesoría), ...

RF-7: Asociación de información bancaria a los clientes

Descripción: Los clientes pueden tener información bancaria asociada a través de un número de cuenta (CCC).

Atributos asociados: idCliente (referencia al cliente), CCC (identificador único del banco), ..

RF-8: Gestión de puntos de suministro de gas N/A

Descripción: Los clientes pueden tener múltiples puntos de suministro de gas (CUPS_gas) asociados, junto con su ubicación.

Atributos asociados: idCliente (referencia al cliente), CUPS_gas (único), ZIP (referencia a la ubicación), …

RF-9: Asignación de empleados a clientes

Descripción: Los empleados pueden estar asignados a clientes específicos.

Atributos asociados: idAgent (identificador único del empleado), codigoempleado, idCliente (referencia al cliente), …

RF-10: Gestión de contratos de clientes

Descripción: El sistema debe permitir la gestión de contratos de clientes, incluyendo información como fechas de inicio y finalización, estado del contrato, etc.

Atributos asociados: idContrato (identificador único del contrato), idCliente (referencia al cliente), fecha_inicio, fecha_fin, estado, …

REGLAS DE NEGOCIO:

RN-1: El titular del contrato debe ser mayor de edad.

Descripción: El sistema debe verificar que la fecha de nacimiento del cliente cumpla con los requisitos de edad mínima para ser titular de un contrato.

Atributos asociados: fecha_nacimiento del Cliente

RN-2: Los números de teléfono deben tener un formato válido.

Descripción: El sistema debe validar que los números de teléfono proporcionados por los clientes sigan un formato específico y sean válidos.

Atributos asociados: telefono del Cliente

RN-3: Los puntos de suministro de energía deben estar asociados a una ubicación válida.

Descripción: El sistema debe garantizar que los puntos de suministro de energía (CUPS_luz) estén asociados a una ubicación (ZIP) válida en la tabla de Ubicacion.

Atributos asociados: CUPS_luz, ZIP en la tabla Cliente y ZIP en la tabla Ubicacion

RN-4: Los puntos de suministro de gas deben estar asociados a una ubicación válida.

Descripción: El sistema debe garantizar que los puntos de suministro CUPS estén asociados a una ubicación (ZIP) válida en la tabla de Ubicacion.

Atributos asociados: CUPS_gas, ZIP en la tabla Cliente y ZIP en la tabla Ubicacion

RN-5: Los empleados asignados a clientes deben tener un código válido.

Descripción: El sistema debe verificar que los empleados asignados a clientes tengan un código válido en la tabla de Empleado.

Atributos asociados: codigoempleado en la tabla Empleado

RN-6: La asignación de empleados a clientes debe ser coherente.

Descripción: El sistema debe verificar que la asignación de empleados a clientes siga reglas específicas establecidas por la empresa, como asignar un número máximo de empleados por cliente.

Atributos asociados: idCliente y idAgent en la tabla Empleado

RN-7: Los números de cuenta bancaria (CCC) deben tener un formato válido.

Descripción: El sistema debe validar que los números de cuenta bancaria proporcionados por los clientes sigan un formato específico y sean válidos.

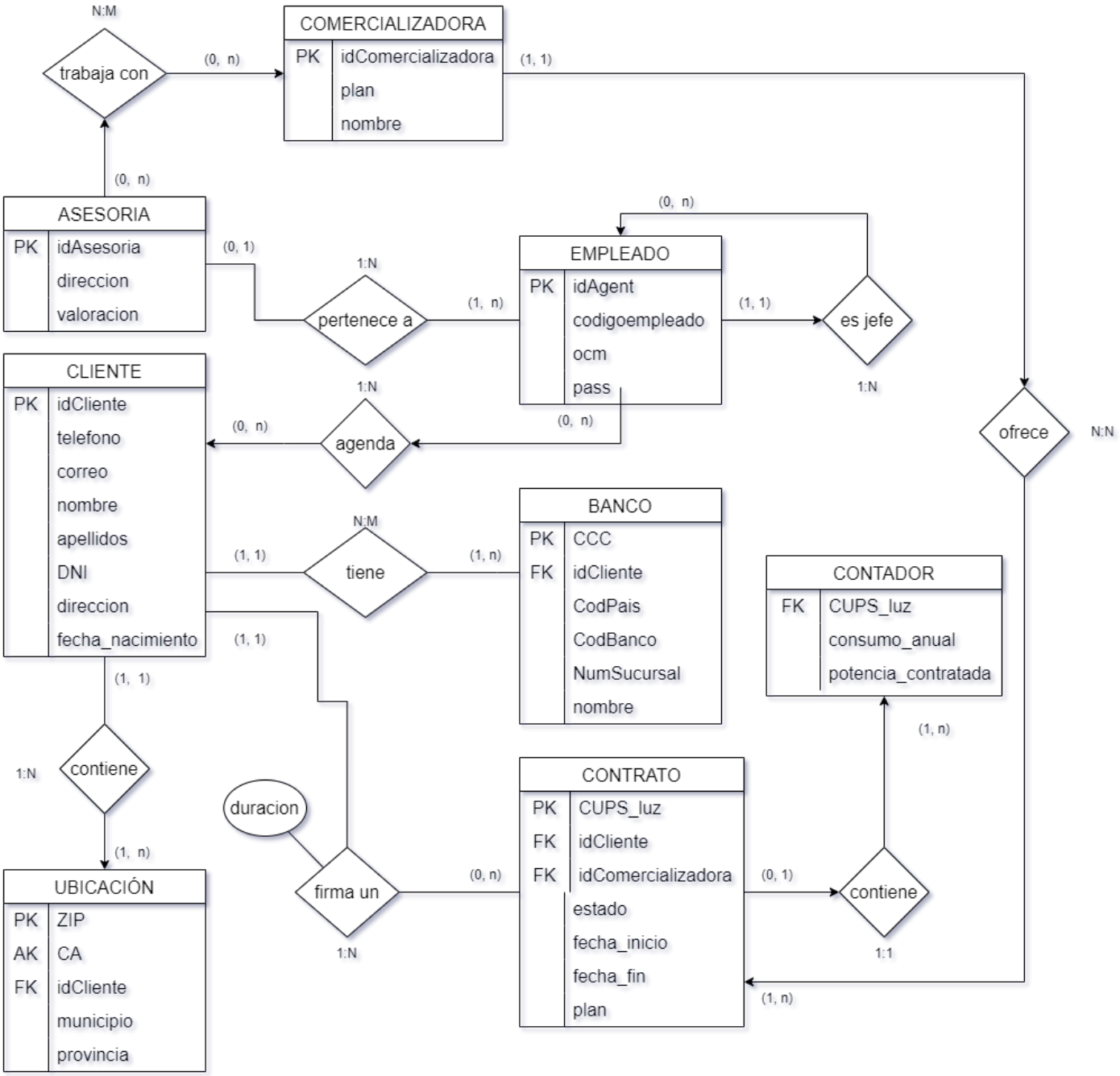
Atributos asociados: CCC en la tabla Cliente

PROYECTO BASE DE DATOS
Álvaro Rodríguez Molina

FASE 2: MODELO ENTIDAD RELACIÓN

ESTRUCTURA PRINCIPAL:

Contador contiene contrato debe ser 1:1



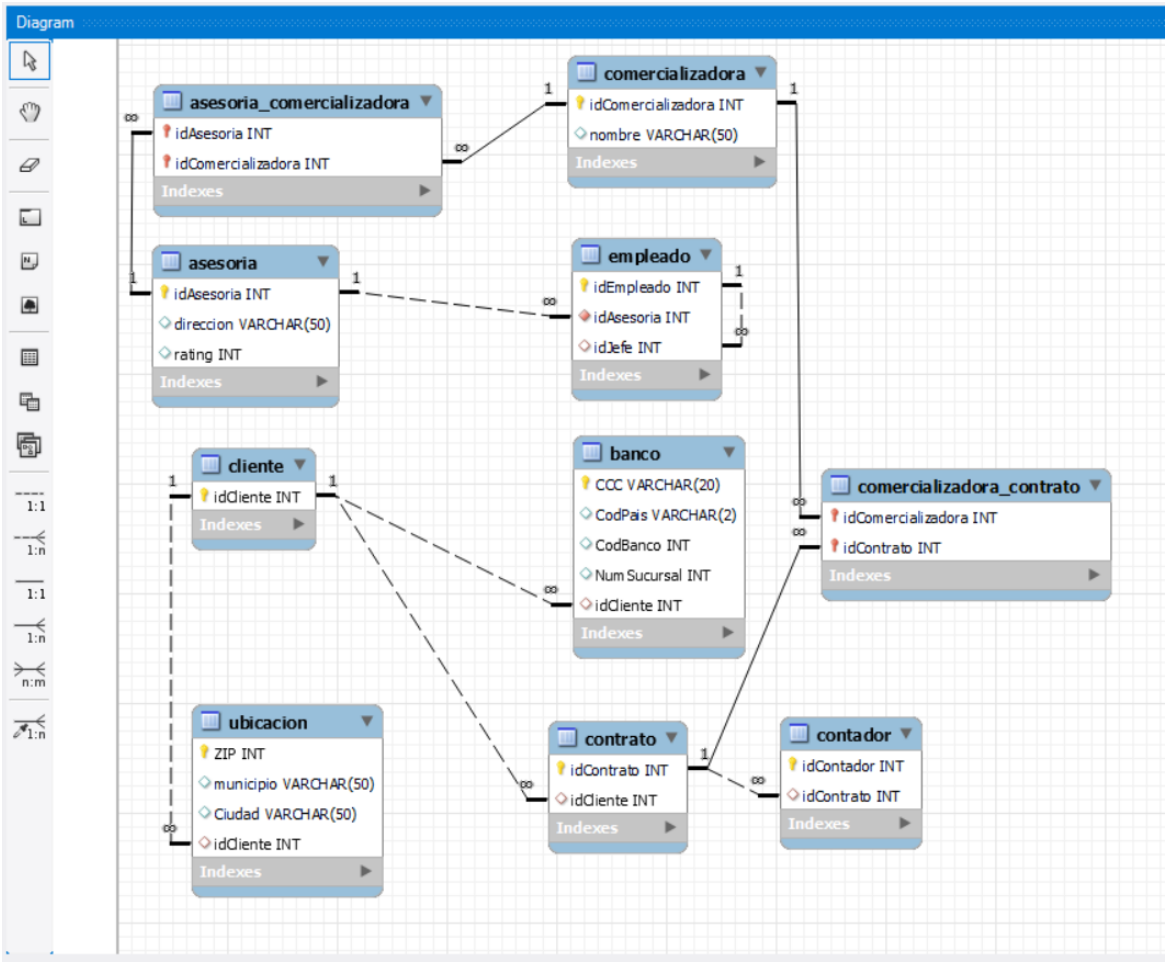
FASE 3: DISEÑO LÓGICO

ESTRUCTURA PRINCIPAL:

ASESORIA trabaja con (N:M) COMERCIALIZADORA
EMPLEADO pertenece a (1:n) ASESORIA
EMPLEADO es jefe (1:n) EMPLEADO
EMPLEADO agenda(1:N) CLIENTE
CLIENTE tiene (1:N) BANCO
CLIENTE contiene (1:N) UBICACION
CLIENTE firma (1:N) CONTRATO
CONTRATO contiene (1:1) CONTADOR
COMERCIALIZADORA ofrece (N:M) CONTRATO

Reconsideraciones del modelo conceptual:

Así quedaría el diseño lógico y por lo tanto la fase 3 del proyecto:

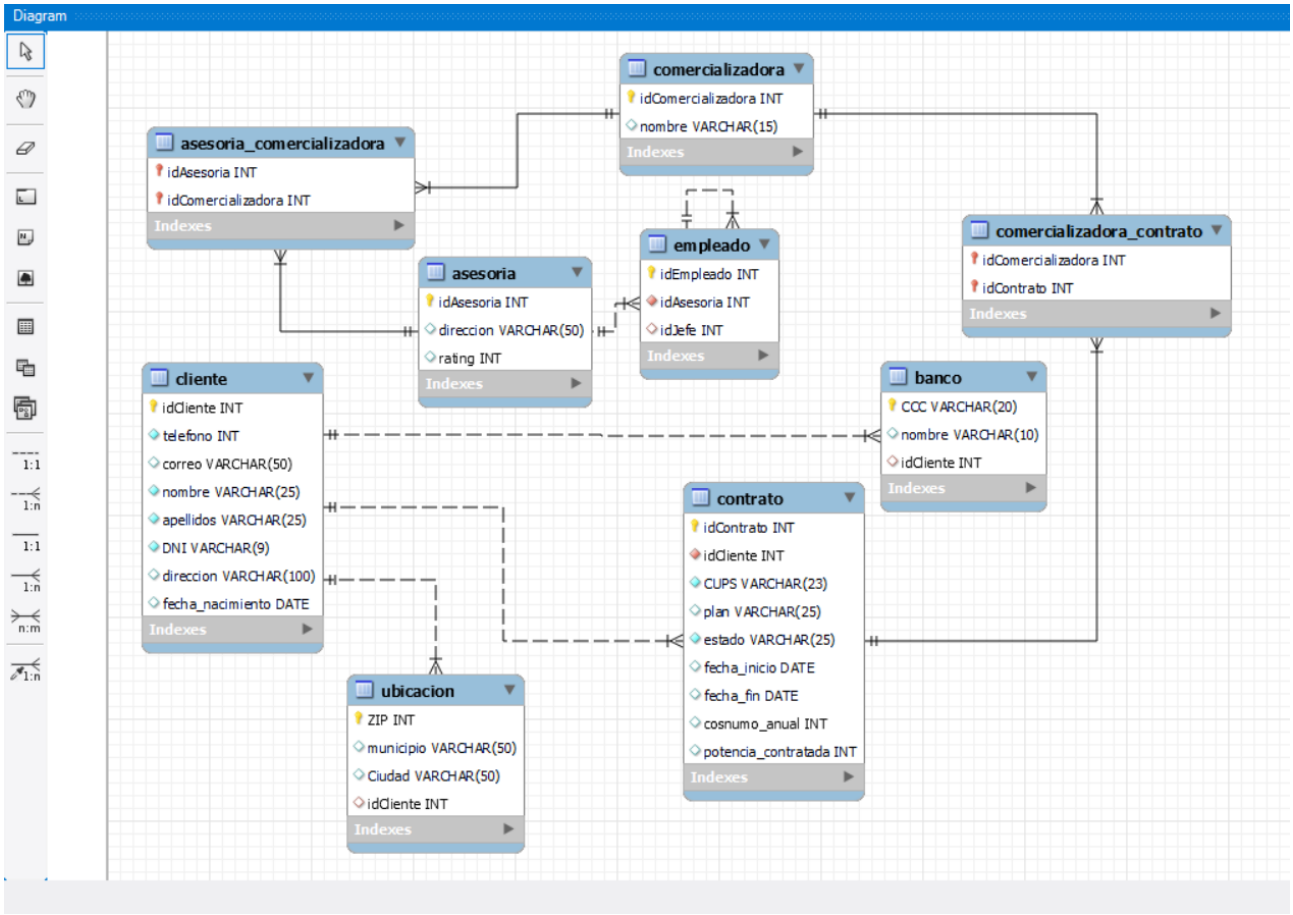


Para crear una base de datos que siga al pie de la letra los diferentes requisitos funcionales y las reglas de negocio anteriormente formalizadas, para esta fase del proyecto he utilizado las propiedades UNIQUE, NOT NULL y las FOREING KEYS correspondientes en las tablas de manera que sigan el modelo entidad relación de la fase 2.

Se han eliminado los atributos: codPais, codBanco y numsucursal de la tabla BANCO, CA de la tabla UBICACION.

Se ha eliminado la tabla CONTADOR, agregando los atributos de la misma a la tabla CONTRATO, la cuál se le ha añadido el atributo Plan, que anteriormente en la FASE 2 correspondía a la tabla COMERCIALIZADORA.

Adjunto la Ingeniería Inversa con el modelo actualizado del modelo físico:



NOTA: La clave primaria de Ubicación se modifica a idUbicacion y el ZIP pasa a ser un atributo más de la misma. A empleado se le ha añadido adicionalmente los atributos: nombre y apellidos.

DEFINICIÓN DE 3 FORMAS NORMALES:

NF 1 o forma normal 1: una tabla que no contiene datos repetidos y que, además, contiene una sola clave única. Un ejemplo podría ser una tabla que tiene un identificador de un cliente (haciendo el papel de la clave única) más un campo que contenga los datos correspondientes al cliente.

NF 2 o forma normal 2: es una tabla que contiene el primer nivel, es decir, forma normal 1, en el que todo registro depende únicamente de la clave principal y la posibilidad de que las columnas puedan depender de otras tablas, pero campos que sean claves en sus respectivas tablas.

NF 3 o forma normal 3: es una tabla que contiene el segundo nivel, es decir, forma normal 2 y que, a su vez, ya está normalizada, porque tiene todas las demás. Estas tablas no pueden tener columnas que dependan de otras columnas que no sean la clave principal.

La últimas formas no son tan detalladas, pues se llega a ellas rápidamente. Estas son las de tipo 4FN, 5FN y la forma normal de dominio/clave, que son las últimas formas normales.

ESTE PROYECTO ESTÁ EN TERCERA FORMA NORMAL

FASE 4: DISEÑO FÍSICO

ESTRUCTURA PRINCIPAL:

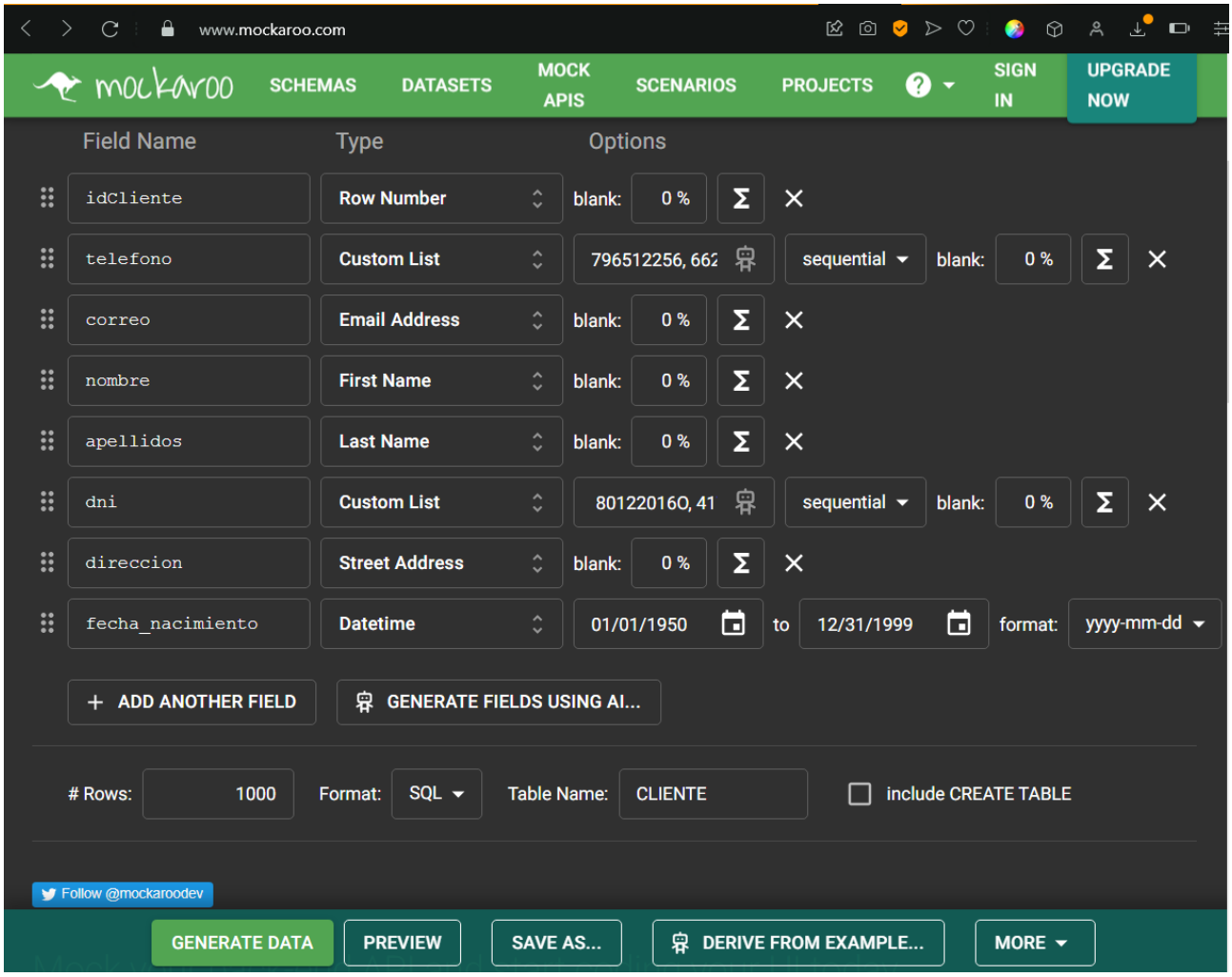
4.1 N/A

4.2 Transformación del modelo lógico en el modelo físico

```sql
DROP DATABASE IF EXISTS StrattonAPP_3;
CREATE DATABASE StrattonAPP_3;
USE StrattonAPP_3;

CREATE TABLE CLIENTE (
  idCliente INT PRIMARY KEY AUTO_INCREMENT,
  telefono INT(9) UNIQUE NOT NULL,
  correo VARCHAR(25) UNIQUE,
  nombre VARCHAR(25),
  apellidos VARCHAR(25),
  DNI VARCHAR(9),
  direccion VARCHAR(100),
  fecha_nacimiento DATE
);

CREATE TABLE BANCO (
  CCC VARCHAR(20) PRIMARY KEY unique,
  nombre VARCHAR(10),
  idCliente INT,
  FOREIGN KEY (idCliente) REFERENCES CLIENTE(idCliente)
);

CREATE TABLE UBICACION (
  idUbicacion INT PRIMARY KEY NOT NULL,
  ZIP INT(5),
  municipio VARCHAR(50),
  Ciudad VARCHAR(50),
  idCliente INT,
  FOREIGN KEY (idCliente) REFERENCES CLIENTE(idCliente)
);

CREATE TABLE COMERCIALIZADORA (
  idComercializadora INT PRIMARY KEY not null,
  nombre VARCHAR(15)
);

CREATE TABLE ASESORIA (
  idAsesoria INT PRIMARY KEY not null,
  direccion VARCHAR(50),
  rating INT
);

CREATE TABLE EMPLEADO (
  idEmpleado INT PRIMARY KEY not null,
  idAsesoria INT not null,
  idJefe INT, -- nombre del jefe, o si es jefe o no
  FOREIGN KEY (idAsesoria) REFERENCES ASESORIA(idAsesoria),
  FOREIGN KEY (idJefe) REFERENCES EMPLEADO(idEmpleado)
);

CREATE TABLE CONTRATO (
  idContrato INT PRIMARY KEY,
  idCliente INT NOT NULL,
  CUPS VARCHAR(23) NOT NULL,
  plan VARCHAR(25),
  estado VARCHAR(25) NOT NULL,
  fecha_inicio DATE,
  fecha_fin DATE, -- TE AVISARÁ A LOS 77 DÍAS
  cosnumo_anual INT,
  potencia_contratada INT,
  FOREIGN KEY (idCliente) REFERENCES CLIENTE(idCliente)
);

CREATE TABLE ASESORIA_COMERCIALIZADORA (
  idAsesoria INT,
  idComercializadora INT,
  FOREIGN KEY (idAsesoria) REFERENCES ASESORIA(idAsesoria),
  FOREIGN KEY (idComercializadora)
  REFERENCES COMERCIALIZADORA(idComercializadora),
  PRIMARY KEY (idAsesoria, idComercializadora)
);

CREATE TABLE COMERCIALIZADORA_CONTRATO (
  idComercializadora INT,
  idContrato INT,
  FOREIGN KEY (idComercializadora)
  REFERENCES COMERCIALIZADORA(idComercializadora),
  FOREIGN KEY (idContrato) REFERENCES CONTRATO(idContrato),
  PRIMARY KEY (idComercializadora, idContrato)
);
```

Para generar datos aleatorios en nuestra base de datos, accedemos a la herramienta gratuita online https://www.mockaroo.com

Para generar los números de teléfono y los dni's únicos, en lugar de utilizar esta herramienta he optado por hacer un pequeño programa en Python bastante simple, que me ha generado los datos para insertarlos directamente en mockaroo y así asegurarme de que los 1000 valores de la tabla cliente estarán correctamente definidos.



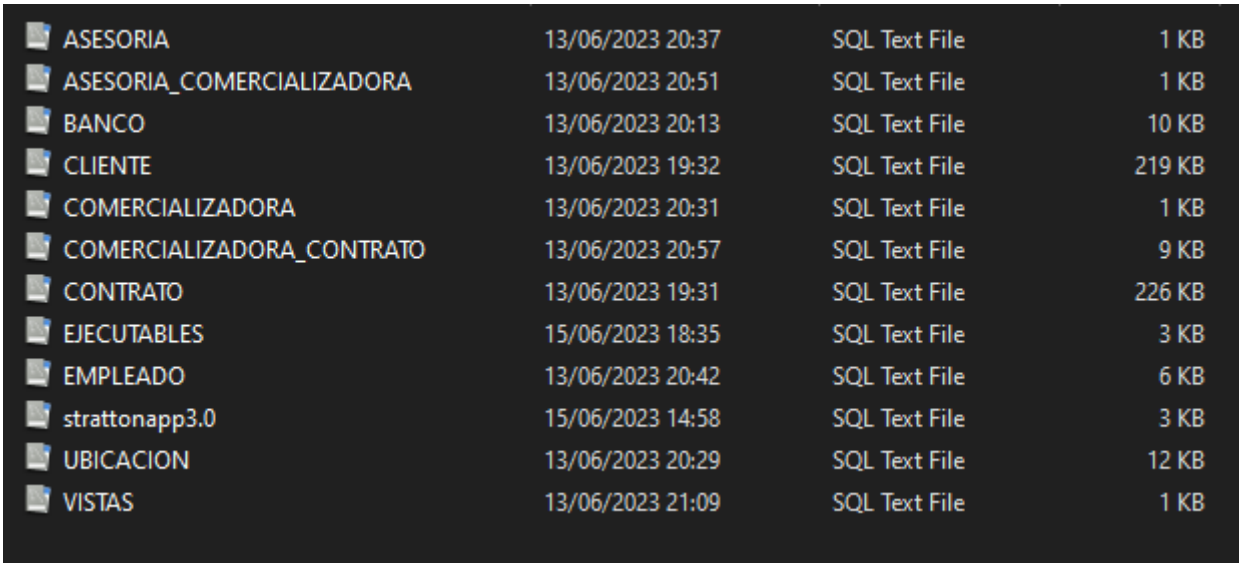

```python
import random

numeros = set()  # Utilizamos un conjunto para evitar duplicados
while len(numeros) < 1000:
    numero = random.randint(600000000, 799999999)  # Genera un número aleatorio entre 600000000 y 799999999
    numeros.add(numero)

numeros_formateados = [str(numero).zfill(9) for numero in numeros]  # Rellena con ceros a la izquierda hasta alcanzar 9 dígitos
numeros_separados = ', '.join(numeros_formateados)  # Une los números separados por comas

print(numeros_separados)
```

PROYECTO BASE DE DATOS
Álvaro Rodríguez Molina

EN LOS INSERTS ES IMPORTANTE SEGUIR UN ORDEN,

1º ASESORÍA → 2º EMPLEADO

1º CLIENTE → 2º CONTRATO



ASESORIA	13/06/2023 20:37	SQL Text File	1 KB
ASESORIA_COMERCIALIZADORA	13/06/2023 20:51	SQL Text File	1 KB
BANCO	13/06/2023 20:13	SQL Text File	10 KB
CLIENTE	13/06/2023 19:32	SQL Text File	219 KB
COMERCIALIZADORA	13/06/2023 20:31	SQL Text File	1 KB
COMERCIALIZADORA_CONTRATO	13/06/2023 20:57	SQL Text File	9 KB
CONTRATO	13/06/2023 19:31	SQL Text File	226 KB
EJECUTABLES	15/06/2023 18:35	SQL Text File	3 KB
EMPLEADO	13/06/2023 20:42	SQL Text File	6 KB
strattonapp3.0	15/06/2023 14:58	SQL Text File	3 KB
UBICACION	13/06/2023 20:29	SQL Text File	12 KB
VISTAS	13/06/2023 21:09	SQL Text File	1 KB

-

La base de datos se llama “stattonapp3.0”

EJECUTABLES” consta de las consultas, los procedimientos y las funciones.

Se adjuntarán los Scripts en el mismo ZIP don de se encuentra este documento.

```
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (1, 796512256, 'dcartlidge0@auda.org.au', 'Danyette', 'Cartlidge',
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (2, 662882305, 'rcanner1@ycombinator.com', 'Ree', 'Canner', '417638
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (3, 658163716, 'mmcdavid2@mashable.com', 'Marietta', 'McDavid', '35
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (4, 622299141, 'egianuzzi3@forbes.com', 'Everard', 'Gianuzzi', '795
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (5, 792410119, 'bhevey4@dmoz.org', 'Benedetta', 'Hevey', '414764291
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (6, 640587784, 'hpoacher5@bandcamp.com', 'Hatty', 'Poacher', '30545
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (7, 743573513, 'emcphaden6@sourceforge.net', 'Eada', 'McPhaden', '1
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (8, 700059658, 'fmatteacci7@patch.com', 'Ferne', 'Matteacci', '6687
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (9, 758487058, 'grisley8@dyndns.org', 'Gypsy', 'Risley', '38717031H
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (10, 780879890, 'rcollopy9@joomla.org', 'Riordan', 'Collopy', '9591
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (11, 657676306, 'apandiea@stumbleupon.com', 'Adair', 'Pandie', '251
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (12, 768897047, 'rgilliverb@ca.gov', 'Roley', 'Gilliver', '38994607
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (13, 716365852, 'ebadenc@patch.com', 'Estrella', 'Baden', '96981832
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (14, 687513633, 'dosgard@cbc.ca', 'Dolores', 'Osgar', '57056013Q',
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (15, 720334889, 'agoncaloe@ucoz.com', 'Amaleta', 'Goncalo', '927868
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (16, 670486570, 'ccorstanf@ow.ly', 'Callean', 'Corstan', '77669614E
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (17, 741697581, 'ufallsg@plala.or.jp', 'Urban', 'Falls', '30625034L
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (18, 615475247, 'spippinh@techcrunch.com', 'Stacia', 'Pippin', '926
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (19, 661194802, 'hmcqueeni@irs.gov', 'Haslett', 'Mc Queen', '593755
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (20, 701337650, 'lclooneyj@virginia.edu', 'Lesya', 'Clooney', '2156
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (21, 654506035, 'ophillipk@vimeo.com', 'Ofella', 'Phillip', '258933
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (22, 666153017, 'bcastell@cdbaby.com', 'Burt', 'Castel', '40294514M
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (23, 726110266, 'tcolyerm@exblog.jp', 'Toddie', 'Colyer', '13084769
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (24, 731600961, 'banandn@mozilla.com', 'Burt', 'Anand', '70524115X'
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (25, 600756289, 'mabbso@tripadvisor.com', 'Matteo', 'Abbs', '273453
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (26, 639903822, 'thugonnetp@webnode.com', 'Therine', 'Hugonnet', '6
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (27, 751102034, 'tnansonq@mac.com', 'Trumann', 'Nanson', '31024217N
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (28, 775340115, 'dsandercroftr@seesaa.net', 'Ddene', 'Sandercroft',
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (29, 719104084, 'adibberts@dell.com', 'Alberik', 'Dibbert', '578987
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (30, 651608149, 'rboikt@last.fm', 'Rooney', 'Boik', '80573167J', '6
insert into CLIENTE (idCliente, telefono, correo, nombre, apellidos, dni, direccion, fecha_nacimiento) values (31, 626487378, 'cboyceu@51.la', 'Cointon', 'Boyce', '40556370V', '
```

FASE 5: IMPLEMENTACIÓN DE LA BASE DE DATOS

ESTRUCTURA PRINCIPAL:

5.1. Procesamiento y optimización de consultas

CONSULTA 1:

Muestra el número total de contratos de cada comercializadora

```
SELECT co.nombre, COUNT(*) AS cantidad_contratos
FROM COMERCIALIZADORA co
INNER JOIN COMERCIALIZADORA_CONTRATO cc ON co.idComercializadora =
cc.idComercializadora
GROUP BY co.nombre;
```

```
31      -- Muestra la cantidad de contratos que tiene cada comercializadora
32 •    SELECT co.nombre, COUNT(*) AS cantidad_contratos
33      FROM COMERCIALIZADORA co
34      INNER JOIN COMERCIALIZADORA_CONTRATO cc ON co.idComercializadora = cc.idComercializadora
35      GROUP BY co.nombre;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	cantidad_contratos
IBERDROLA	39
NATURGY	34
ENDESA	27

CONSULTA 2:

Muestra los clientes con contrato Inactivo, para su renovación.

```
SELECT c.nombre, c.apellidos, c.telefono, ct.idContrato, ct.estado
FROM CLIENTE c
INNER JOIN CONTRATO ct ON c.idCliente = ct.idCliente
WHERE ct.estado = 'Inactivo';
```

```
37      -- Muestra los clientes inactivos en el parner
38 •    SELECT c.nombre, c.apellidos, c.telefono, ct.idContrato, ct.estado
39      FROM CLIENTE c
40      INNER JOIN CONTRATO ct ON c.idCliente = ct.idCliente
41      WHERE ct.estado = 'Inactivo';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	apellidos	telefono	idContrato	estado
Urban	Falls	741697581	17	Inactivo
Haslett	Mc Queen	661194802	19	Inactivo
Ofella	Phillip	654506035	21	Inactivo
Toddie	Colyer	726110266	23	Inactivo
Burt	Anand	731600961	24	Inactivo
Erna	Lukianovich	640727127	32	Inactivo
Calley	Somerton	799606875	36	Inactivo
Melisent	Neles	631980126	38	Inactivo
Francklyn	Lecordier	773652574	41	Inactivo
Camile	Goldstraw	643856483	42	Inactivo
Brunhilde	Treagust	754597995	46	Inactivo
Lane	Gales	784095346	48	Inactivo
Davida	Goublier	620675199	55	Inactivo
Lenee	Talman	737349761	57	Inactivo
Fitzgerald	Bertomeu	628066437	59	Inactivo

CONSULTA 3:

Muestra los suministros con una potencia contratada mayor, se utilizará para determinar que contratos son 2.1 y 3.0 de mayor potencia

```
SELECT *
FROM CONTRATO
WHERE potencia_contratada > 5000;
```

```
49 •  SELECT *
50    FROM CONTRATO
51    WHERE potencia_contratada > 5000;
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

idContrato	idCliente	CUPS	plan	estado	fecha_inicio	fecha_fin	cosnumo_anual	potencia_contratada
2	2	ES0031126182946755SX	Basic	Pendiente	2011-06-22	2019-04-03	316518	7613
3	3	ES0031135646310573LO	Pro	Pendiente	2014-06-08	2016-11-14	465036	6126
4	4	ES0031129339229327ZX	Premium	Pendiente	2021-04-03	2011-06-15	537266	6977
6	6	ES0031161061528482RQ	Premium	Activo	2017-05-08	2008-02-15	762360	9989
7	7	ES0031134270560410QJ	Pro	Pendiente	2004-12-01	2006-02-12	80535	6130
8	8	ES0031110658082593GO	Premium	Pendiente	2004-01-14	2012-09-27	486362	5329
9	9	ES0031180153399884HR	Pro	Activo	2004-07-30	2007-04-27	569341	7646
10	10	ES0031187106124929EM	Basic	Pendiente	2003-08-18	2018-06-13	715919	8803
11	11	ES0031143937406879GV	Premium	Pendiente	2013-12-06	2017-02-12	217007	7827
12	12	ES0031118970439024DO	Premium	Activo	2019-03-31	2010-12-05	414622	5249
14	14	ES0031169865530227MX	Basic	Activo	2019-10-26	2019-06-15	111848	9793
15	15	ES0031164636922596WF	Basic	Pendiente	2018-06-13	2006-06-18	725358	8612
17	17	ES0031144851804101GP	Pro	Inactivo	2001-12-27	2001-06-30	158210	7870
18	18	ES0031187689080567JY	Premium	Activo	2005-01-07	2008-02-22	308216	6714
19	19	ES0031195451092513DP	Pro	Inactivo	2018-08-13	2015-12-01	305023	9400
20	20	ES0031118981878404KX	Premium	Activo	2021-11-29	2002-07-02	344092	5560
21	21	ES0031124140201498TL	Basic	Inactivo	2011-02-28	2005-08-07	480665	5519
22	22	ES0031165810266557EA	Basic	Pendiente	2016-06-21	2006-08-17	85375	5557
23	23	ES0031163711427471PQ	Pro	Inactivo	2007-02-09	2019-05-09	238929	8696
24	24	ES0031126871935794DC	Premium	Inactivo	2014-11-24	2004-09-03	169461	8832

-

CONSULTA 4:

Muestra el número de clientes por provincia

```
SELECT u.Ciudad, COUNT(c.idCliente) AS numero_clientes
FROM UBICACION u
INNER JOIN CLIENTE c ON u.idCliente = c.idCliente
GROUP BY u.Ciudad;
```

```
52      -- Numero de clientes por provincia
53 •    SELECT u.Ciudad, COUNT(c.idCliente) AS numero_clientes
54      FROM UBICACION u
55      INNER JOIN CLIENTE c ON u.idCliente = c.idCliente
56      GROUP BY u.Ciudad;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Ciudad	numero_clientes
Sevilla	13
Málaga	13
Cádiz	13
Jaen	13
Granada	12
Córdoba	12
Almería	12
Huelva	12

CONSULTA 5:

Muestra el número de cuentas que hay en cada banco

```
SELECT BANCO.nombre, COUNT(BANCO.CCC) AS num_cuentas
FROM BANCO
GROUP BY BANCO.nombre
ORDER BY num_cuentas;
```

```
60 •    SELECT BANCO.nombre, COUNT(BANCO.CCC) AS num_cuentas
61      FROM BANCO
62      GROUP BY BANCO.nombre
63      ORDER BY num_cuentas;
64
```

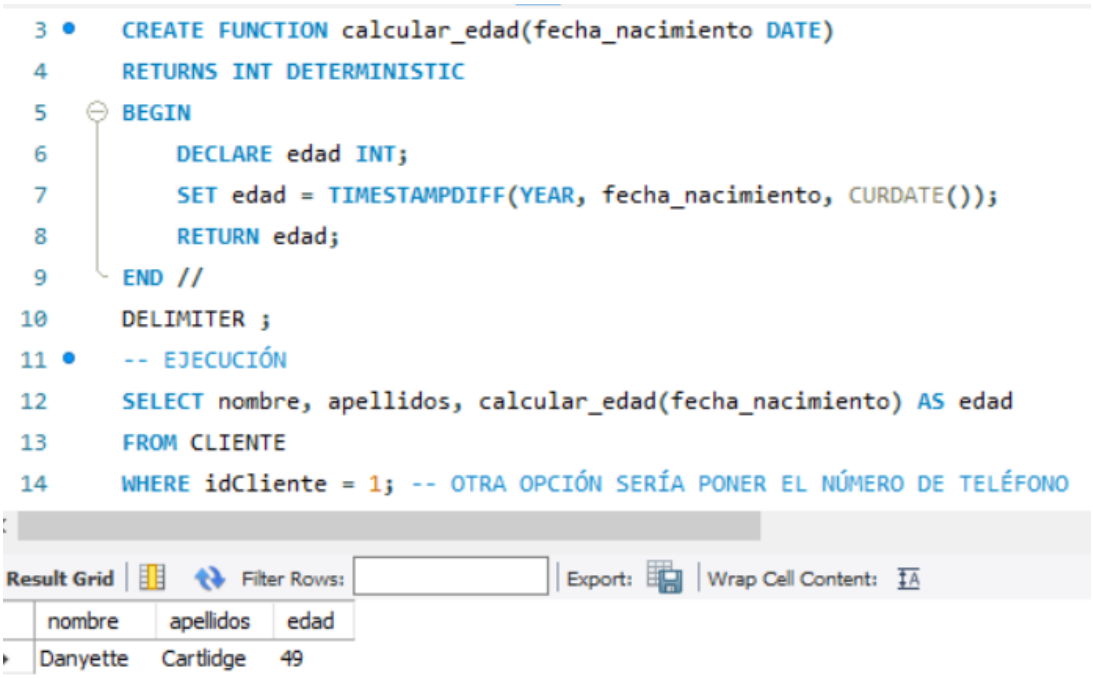
esult Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	num_cuentas
CAIXABANK	28
BBVA	34
SANTANDER	38

FUNCIÓN CALCULAR_EDAD:

Esta función determina la edad de cada cliente introduciendo su idCliente (la idea principal es poner su número de teléfono) para determinar si puede hacer la contratación con una comercializadora o con otra, ya que hay algunas que tienen restricciones de edad.

```
DELIMITER //
CREATE FUNCTION calcular_edad(fecha_nacimiento DATE)
RETURNS INT DETERMINISTIC
BEGIN
    DECLARE edad INT;
    SET edad = TIMESTAMPDIFF(YEAR, fecha_nacimiento, CURDATE());
    RETURN edad;
END //
DELIMITER ;
-- EJECUCIÓN
SELECT nombre, apellidos, calcular_edad(fecha_nacimiento) AS edad
FROM CLIENTE
WHERE idCliente = 1;
```



PROCEDIMIENTO CLIENTES_SIN_RETRO

Este procedimiento indica qué clientes son aptos para renovar su contrato sin que entre retro comisión al empleado de la asesoría.

```
DELIMITER //
CREATE PROCEDURE clientes_sin_retro()
BEGIN
    DECLARE fecha_limite DATE;
    -- Calcula la fecha límite 77 días en el pasado
    SET fecha_limite = DATE_SUB(CURDATE(), INTERVAL 77 DAY);
    -- Obtiene los clientes que no darán retro comisión
    SELECT c.nombre, c.apellidos
    FROM CLIENTE c
    INNER JOIN CONTRATO co ON c.idCliente = co.idCliente
    WHERE co.fecha_inicio <= fecha_limite;
END //
DELIMITER ;
CALL  clientes_sin_retro()
```

```
16      DELIMITER //
17  •   CREATE PROCEDURE clientes_sin_retro()
18      BEGIN
19          DECLARE fecha_limite DATE;
20          -- Calcula la fecha límite 77 días en el pasado
21          SET fecha_limite = DATE_SUB(CURDATE(), INTERVAL 77 DAY);
22          -- Obtiene los clientes que no darán retro comisión
23          SELECT c.nombre, c.apellidos
24          FROM CLIENTE c
25          INNER JOIN CONTRATO co ON c.idCliente = co.idCliente
26          WHERE co.fecha_inicio <= fecha_limite;
27      END //
28      DELIMITER ;
29  •   CALL  clientes_sin_retro()
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

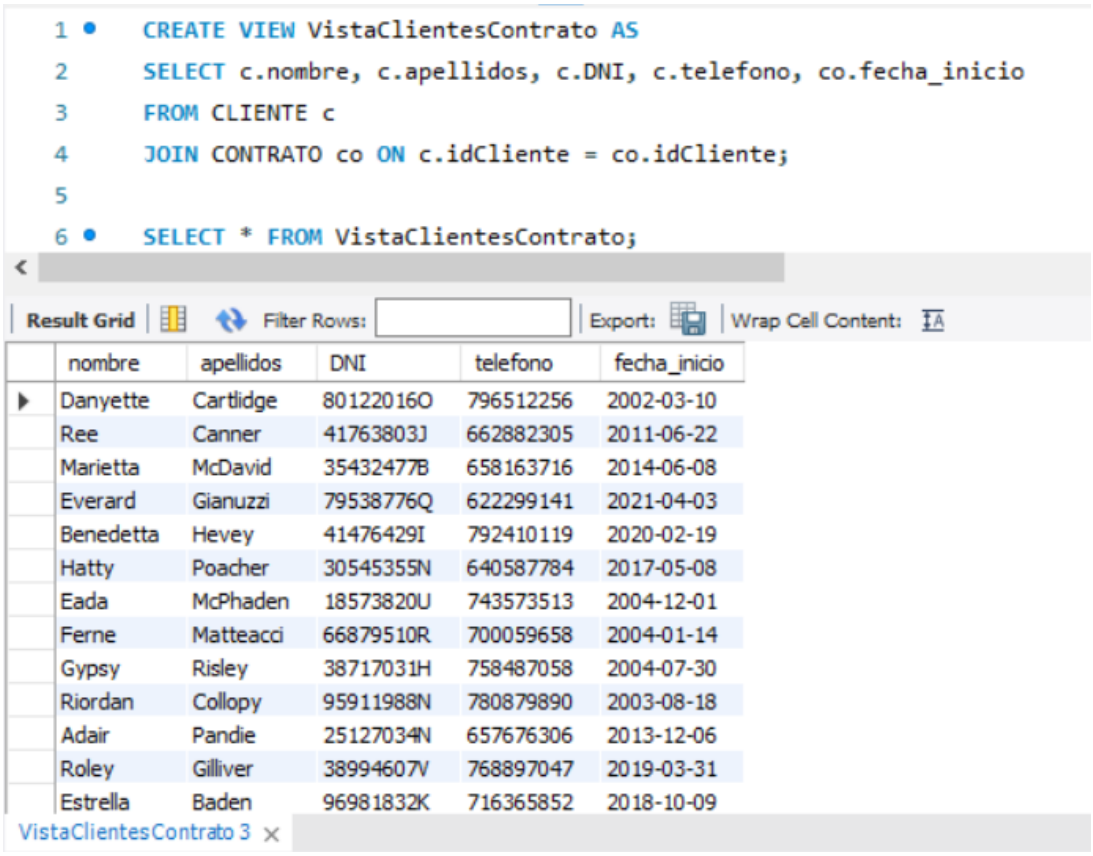
nombre	apellidos	telefono
Danyette	Cartlidge	796512256
Ree	Canner	662882305
Marietta	McDavid	658163716
Everard	Gianuzzi	622299141
Benedetta	Hevey	792410119
Hatty	Poacher	640587784
Eada	McPhaden	743573513
Ferne	Matteacci	700059658
Gypsy	Risley	758487058
Riordan	Collopy	780879890
Adair	Pandie	657676306
Roley	Gilliver	768897047
Estrella	Baden	716365852
Dolores	Osgar	687513633
Amaleta	Goncalo	720334889
Callean	Corstan	670486570

5.2. Procesamiento de vistas

PRIMERA VISTA:

Se desea recoger la fecha de inicio de la contratación de los clientes para su posterior renovación

```
CREATE VIEW VistaClientesContrato AS
SELECT c.nombre, c.apellidos, c.DNI, c.telefono, co.fecha_inicio
FROM CLIENTE c
JOIN CONTRATO co ON c.idCliente = co.idCliente;

SELECT * FROM VistaClientesContrato;
```



```
1 •  CREATE VIEW VistaClientesContrato AS
2    SELECT c.nombre, c.apellidos, c.DNI, c.telefono, co.fecha_inicio
3    FROM CLIENTE c
4    JOIN CONTRATO co ON c.idCliente = co.idCliente;
5
6 •  SELECT * FROM VistaClientesContrato;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	apellidos	DNI	telefono	fecha_inicio
Danyette	Cartlidge	80122016O	796512256	2002-03-10
Ree	Canner	41763803J	662882305	2011-06-22
Marietta	McDavid	35432477B	658163716	2014-06-08
Everard	Gianuzzi	79538776Q	622299141	2021-04-03
Benedetta	Hevey	41476429I	792410119	2020-02-19
Hatty	Poacher	30545355N	640587784	2017-05-08
Eada	McPhaden	18573820U	743573513	2004-12-01
Ferne	Matteacci	66879510R	700059658	2004-01-14
Gypsy	Risley	38717031H	758487058	2004-07-30
Riordan	Collopy	95911988N	780879890	2003-08-18
Adair	Pandie	25127034N	657676306	2013-12-06
Roley	Gilliver	38994607V	768897047	2019-03-31
Estrella	Baden	96981832K	716365852	2018-10-09

VistaClientesContrato 3

SEGUNDA VISTA:

Se desea recoger el CUPS y Número de cuenta del cliente(CCC) junto con sus datos para realizar una contratación.

```
CREATE VIEW VistaClientesCUPS_CCC AS
SELECT c.nombre, c.apellidos, c.DNI, c.telefono, co.CUPS, b.CCC
FROM CLIENTE c
JOIN CONTRATO co ON c.idCliente = co.idCliente
JOIN BANCO b ON c.idCliente = b.idCliente;

SELECT * FROM VistaClientesCUPS_CCC;
```

```
 8 •   CREATE VIEW VistaClientesCUPS_CCC AS
 9     SELECT c.nombre, c.apellidos, c.DNI, c.telefono, co.CUPS, b.CCC
10     FROM CLIENTE c
11     JOIN CONTRATO co ON c.idCliente = co.idCliente
12     JOIN BANCO b ON c.idCliente = b.idCliente;
13
14 •   SELECT * FROM VistaClientesCUPS_CCC;
15
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

nombre	apellidos	DNI	telefono	CUPS	CCC
Danyette	Cartlidge	80122016O	796512256	ES0031148524454906XU	ES4822374962446612360375
Ree	Canner	41763803J	662882305	ES0031126182946755SX	ES9675170859925134982244
Marietta	McDavid	35432477B	658163716	ES0031135646310573LO	ES0178571103026542424527
Everard	Gianuzzi	79538776Q	622299141	ES0031129339229327ZX	ES1102594823563626425649
Benedetta	Hevey	41476429I	792410119	ES0031154336594647ZP	ES8773934691465271027269
Hatty	Poacher	30545355N	640587784	ES0031161061528482RQ	ES0461942857279323038863
Eada	McPhaden	18573820U	743573513	ES0031134270560410QJ	ES8099352280460012047410
Ferne	Matteacci	66879510R	700059658	ES0031110658082593GO	ES3117795150344783564656
Gypsy	Risley	38717031H	758487058	ES0031180153399884HR	ES1169945807304283030226
Riordan	Collopy	95911988N	780879890	ES0031187106124929EM	ES7009310478359343938389
Adair	Randie	25127034N	657676306	ES0031143937406879GV	ES5727325707056948768455

5.3. Administración de la seguridad

N/A

Revisión	Modificación	Fecha
0	Creación del documento	09/01/23
1	Continuación del documento	15/01/23
2	Actualización	16/01/23
3	Actualización 1.0	12/02/2023
4	Actualización 2.0	03/05/2023
5	Actualización 3.0 Aquí empieza el proyecto de verdad, debido a la recuperación del proyecto en la asignatura de Base de Datos, con lo cuál se han considerado cambios en todo el documento y en la estructura principal del proyecto, para aportar mayor solidez y calidad al producto final deseado.	01/06/2023
6	Finalización del proyecto y envío del mismo al profesor	15/06/2023