

BASES DE DATOS 2

MEMORIA DE LA PRÁCTICA 3
CURSO 2017-2018

CARLOS MARAÑES NUENO 717788@CELES.UNIZAR.ES

NICOLÁS LERA LÓPEZ 721808@CELES.UNIZAR.ES



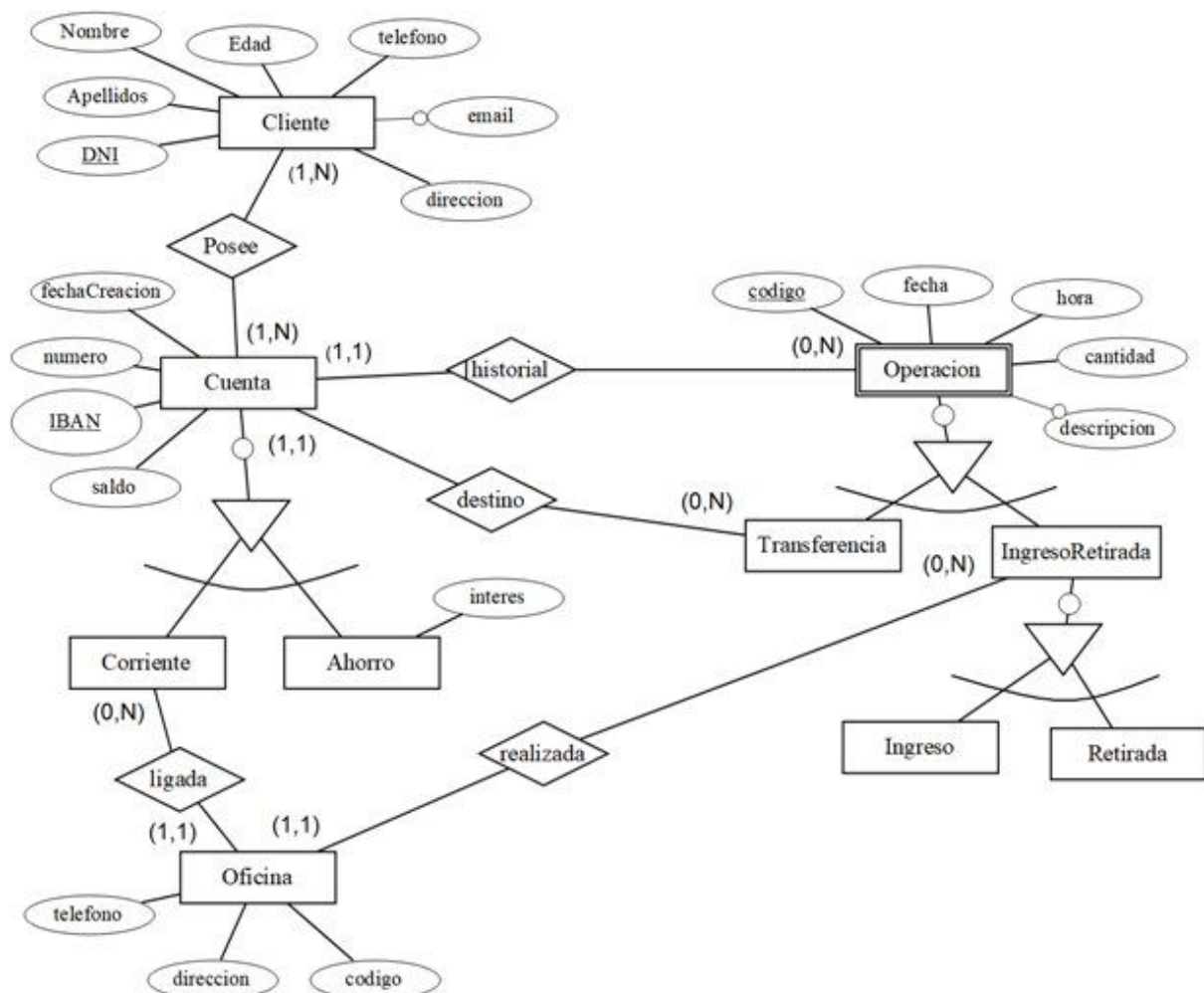
ÍNDICE

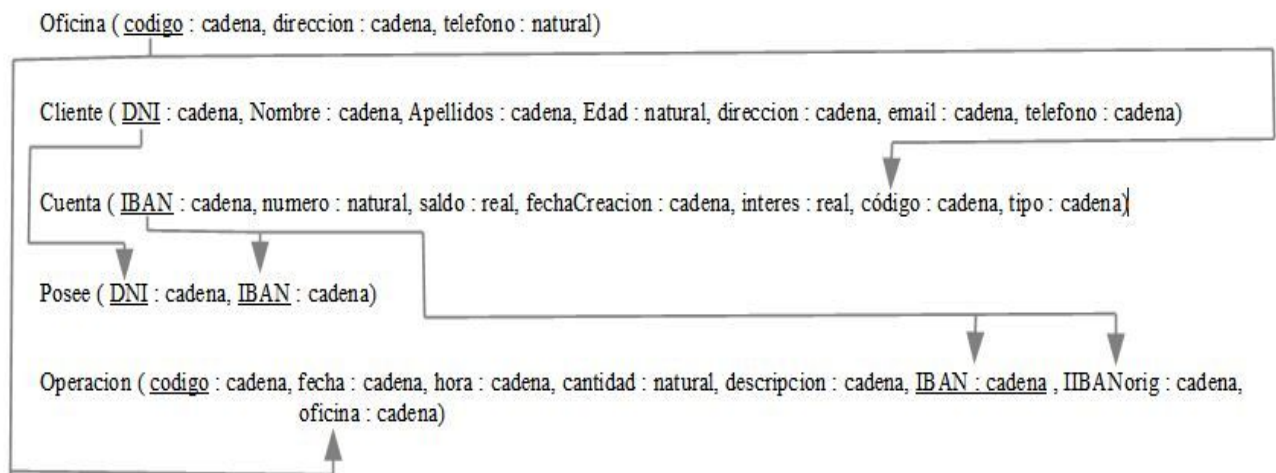
INTRODUCCIÓN	2
PARTE 1 - ESQUEMA CONCEPTUAL Y LÓGICO DE LA BASE DE DATOS RELACIONAL DISEÑADA EN LA PRÁCTICA ANTERIOR	2
PARTE 1 - DETERMINACIÓN DEL ESQUEMA LÓGICO Y CONCEPTUAL DE LA BASE DE DATOS ORACLE A INTEGRAR	3
PARTE 1 - MEJORAS SUGERIDAS PARA LA BASE DE DATOS A INTEGRAR	6
PARTE 1 - DEFINICIÓN E IMPLEMENTACIÓN DEL ESQUEMA GLOBAL EN ORACLE	7
PARTE 2 - ENUNCIADO DE UN PROBLEMA DE DISEÑO DE BASE DE DATOS	14
PARTE 2 - ESQUEMA CONCEPTUAL PARA EL PROBLEMA ENUNCIADO	14
PARTE 2 - ESQUEMA LÓGICO 1 PARA EL PROBLEMA ENUNCIADO	15
PARTE 2 - ESQUEMA LÓGICO 2 PARA EL PROBLEMA ENUNCIADO	15
PARTE 2 - ESQUEMA GLOBAL Y SU IMPLEMENTACIÓN EN POSTGRESQL	16
ACTUALIZACIONES DE DATOS SOBRE EL ESQUEMA GLOBAL	17
DIFICULTADES ENCONTRADAS	23
ESFUERZOS INVERTIDOS	23
BIBLIOGRAFÍA	23

1. INTRODUCCIÓN

En la siguiente práctica se va a llevar a cabo un ejemplo de integración de base de datos federadas para Oracle y PostgreSQL. En el caso de Oracle se integra una base de datos ya proporcionada y la realizada en la práctica anterior. Para PostgreSQL se ha creado un dominio del problema distinto.

2. PARTE 1 - ESQUEMA CONCEPTUAL Y LÓGICO DE LA BASE DE DATOS RELACIONAL DISEÑADA EN LA PRÁCTICA ANTERIOR





En el esquema anterior, hay un problema de diseño: se guarda la edad del cliente en vez de la fecha de nacimiento, por lo que tendría que ser constantemente recalculada. Sin embargo, se ha lidiado con este problema para crear la base federada.

3. PARTE 1 - DETERMINACIÓN DEL ESQUEMA LÓGICO Y CONCEPTUAL DE LA BASE DE DATOS ORACLE A INTEGRAR

Para extraer el esquema lógico y conceptual se ha realizado el siguiente procedimiento:

Para saber los nombres de las tablas:

```
select table_name from user_tables@schema2bd2;
```

Una vez se tienen los nombres de las tablas, se utiliza el comando:

```
desc <nombreTabla>@schema2bd2;
```

De esta manera se saben los tipos de datos, sus nombres y si pueden ser null o no.

El siguiente paso ha sido extraer las relaciones como claves primarias y claves ajenas. Se ha utilizado el script sql disponible en Moodle para realizar su conversión a csv y así poder leer con facilidad los datos.

```
select * from user_constraints@schema2bd2;
```

Con la consulta anterior, se conoce el tipo de constraint, que puede ser:

- P: primary key
- U: unique
- R: references
- C: check

En el campo seach_condition indica si el atributo puede ser null o no.

Con la consulta: `select * from user_cons_columns@schema2bd2;` se pueden traducir las constraints a columnas y así saber qué hacer referencia a qué. Por ejemplo:

	A	B	C	D	E	F	G	H	I
1	OWNER	CONSTRAINT	C	TABLE_NAME	SEARCH_CONDITION	R_OWNER	R_CONSTRAINT	DELETE_RULE	STATUS
2									
3	TEST	SYS_C001402	C	CODENTIDAD	BANCO IS NOT NULL				ENABLED
4	TEST	SYS_C001402	C	CODENTIDAD	CODIGO IS NOT NULL				ENABLED
5	TEST	CODBANCOS	P	CODENTIDAD					ENABLED
6	TEST	SYS_C001402	C	CODPOSTAL	CALLE IS NOT NULL				ENABLED
7	TEST	SYS_C001402	C	CODPOSTAL	CIUDAD IS NOT NULL				ENABLED
8	TEST	SYS_C001402	C	CODPOSTAL	CODPOSTAL IS NOT NULL				ENABLED
9	TEST	CODPOSTAL	P	CODPOSTAL					ENABLED
10	TEST	SYS_C001402	C	CUENTA	CCC IS NOT NULL				ENABLED
11	TEST	SYS_C001402	C	CUENTA	FECHACREACION IS NOT NULL				ENABLED
12	TEST	SYS_C001402	C	CUENTA	SALDO IS NOT NULL				ENABLED
13	TEST	SYS_C001402	C	CUENTA	TITULAR IS NOT NULL				ENABLED
14	TEST	CUENTAPK	P	CUENTA					ENABLED
15	TEST	CUENTATITU	R	CUENTA		TEST	TITULARPK	NO ACTION	ENABLED
16	TEST	SYS_C001402	C	CUENTAHO	TIPOINTERES IS NOT NULL				ENABLED
17	TEST	SYS_C001402	P	CUENTAHO					ENABLED
18	TEST	CUENTAHO	R	CUENTAHO		TEST	CUENTAPK	NO ACTION	ENABLED
19	TEST	SYS_C001402	C	CUENTACOR	SUCURSAL_CODOFICINA IS NOT NULL				ENABLED
20	TEST	SYS_C001402	P	CUENTACOR					ENABLED
21	TEST	CUENTACOR	R	CUENTACOR		TEST	CUENTAPK	NO ACTION	ENABLED
22	TEST	CUENTACOR	R	CUENTACOR		TEST	SYS_C001402	NO ACTION	ENABLED

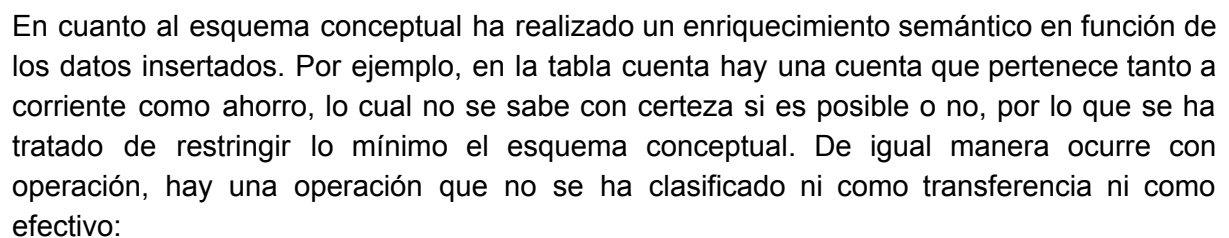
Cuentatitular tiene una constraint de referencia, indicada por la R, en la tabla cuenta e indica que el nombre es titularpk. Ahora se mira en la otra tabla para traducirlo:

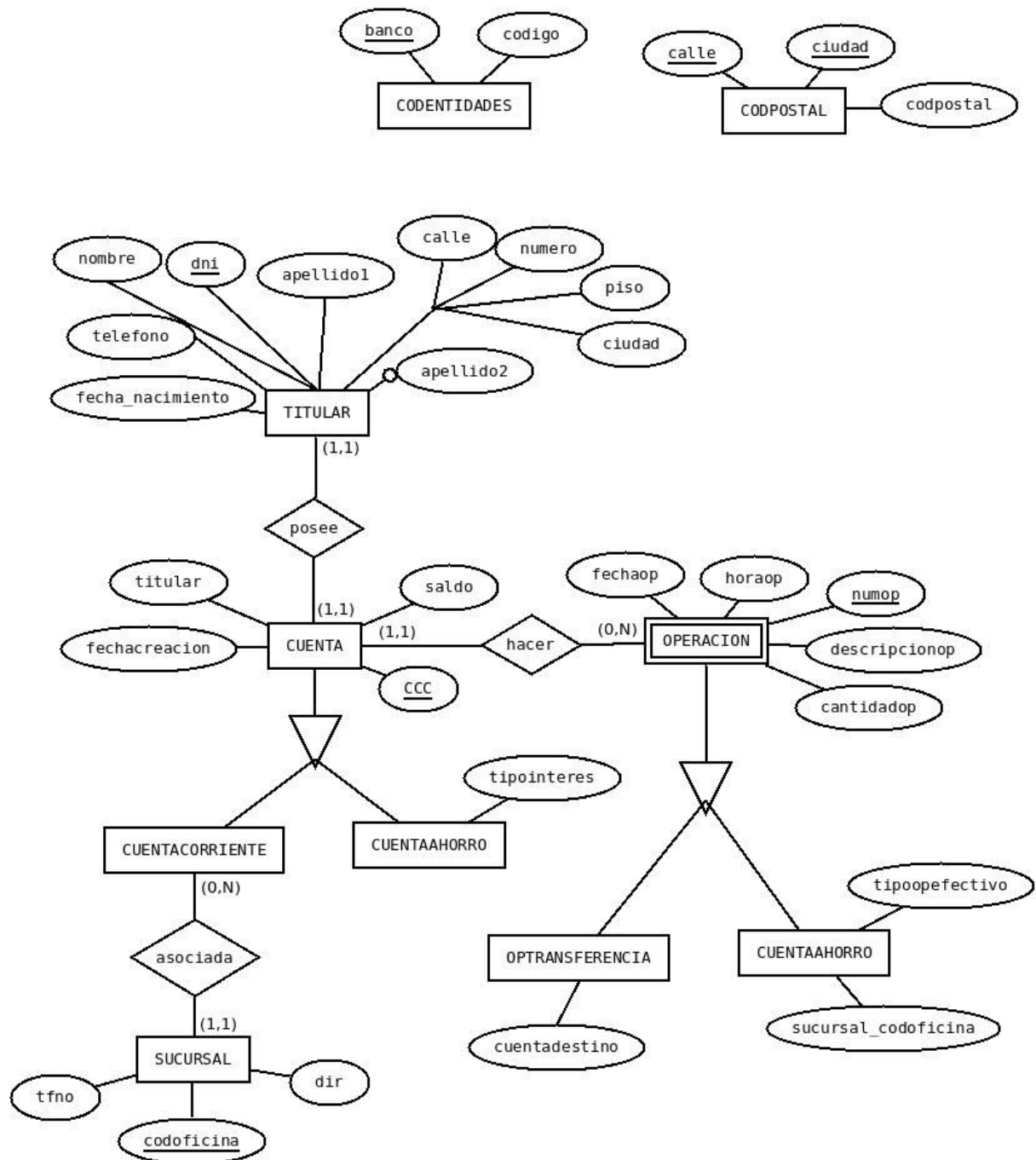
	A	B	C	D	E
1	OWNER	CONSTRAINT	TABLE_NAME	COLUMN_NAME	POSITION
2	TEST	TITULARPK	TITULAR	DNI	1
3	TEST	CODBANCOS	CODENTIDAD	BANCO	1
4	TEST	CODPOSTAL	CODPOSTAL	CALLE	1
5	TEST	CODPOSTAL	CODPOSTAL	CIUDAD	2
6	TEST	CUENTAHO	CUENTAHO	CCC	1
7	TEST	CUENTACOR	CUENTACOR	SUCURSAL_C	1
8	TEST	CUENTACOR	CUENTACOR	CCC	1
9	TEST	CUENTAPK	CUENTA	CCC	1
10	TEST	CUENTATITU	CUENTA	TITULAR	1

Y dice que titularpk se corresponde con el campo DNI de la tabla titular.

Un problema que surge es que las cardinalidades hay que inferirlas a partir de los datos, por lo que se hacen consultas y se comprueba si el mismo elemento en la tabla está repetido pero cambia únicamente su clave ajena, entonces se confirma que la cardinalidad no es 1. Sin embargo, si la clave ajena no es primary key, entonces la cardinalidad debe ser 1 ya que limitarían la repetición de tuplas.

El esquema lógico queda de la siguiente manera:





4. PARTE 1 - MEJORAS SUGERIDAS PARA LA BASE DE DATOS A INTEGRAR

La principal mejora que se observa a simple vista es la eliminación de las tablas Codentidades y Codpostal ya que permanecen aisladas. Sería interesante mantenerlas si, por ejemplo, Codentidades se relaciona con las cuentas y CodPostal con las direcciones, pero no es el caso.

Existen los siguientes problemas de diseño:

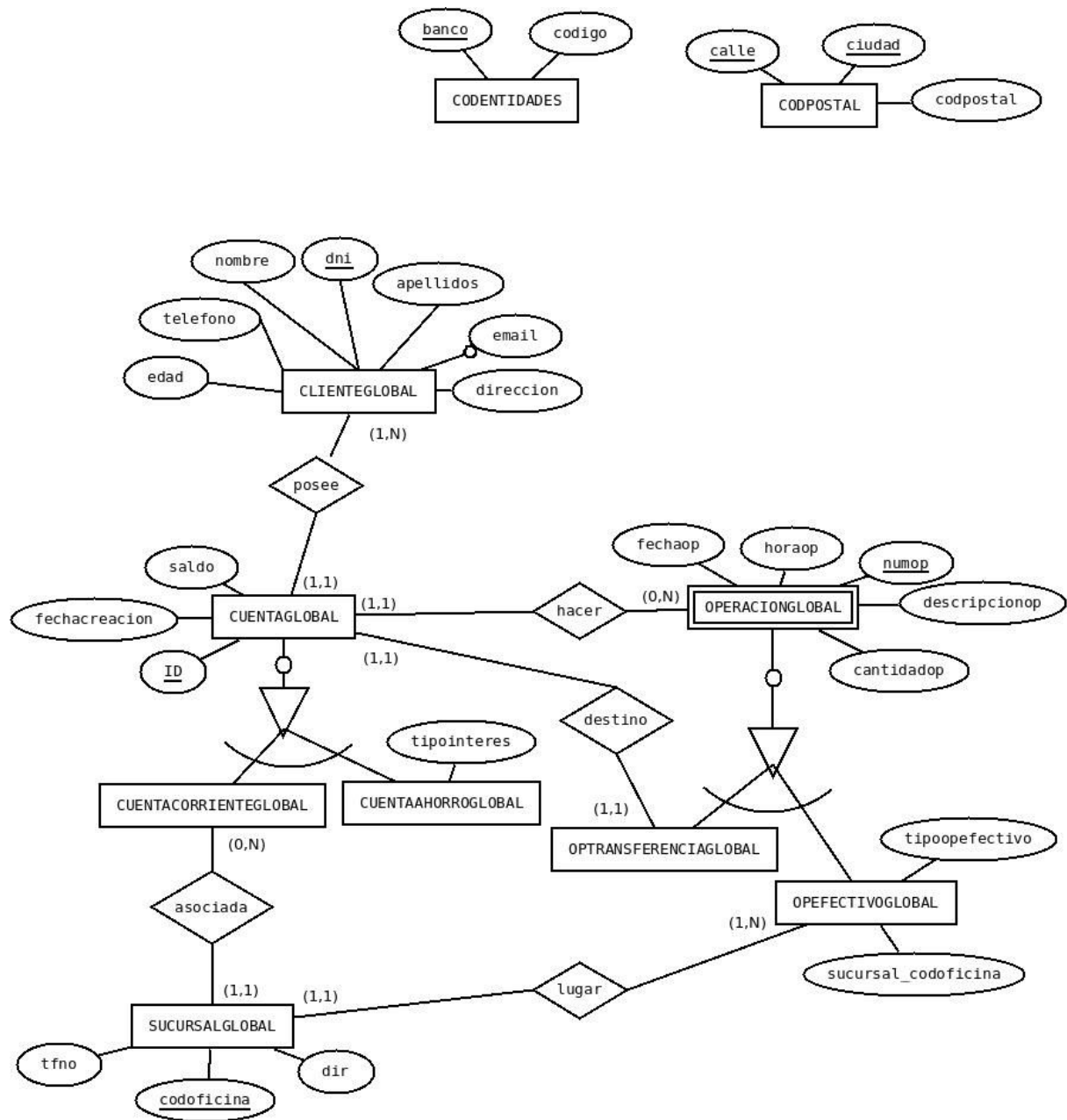
- En OPTRANSFERENCIA, la cuenta de destino no se relaciona con ninguna cuenta, es decir, no tiene restricción de clave ajena.
- En OPEFECTIVO, sucursal_codofina no tiene restricción de clave ajena con oficina.

Para solucionar estos problemas debería existir algún trigger para garantizar la consistencia, pero con la consulta: *select * from user_triggers@schema2bd2*; se puede observar que no existe ningún trigger.

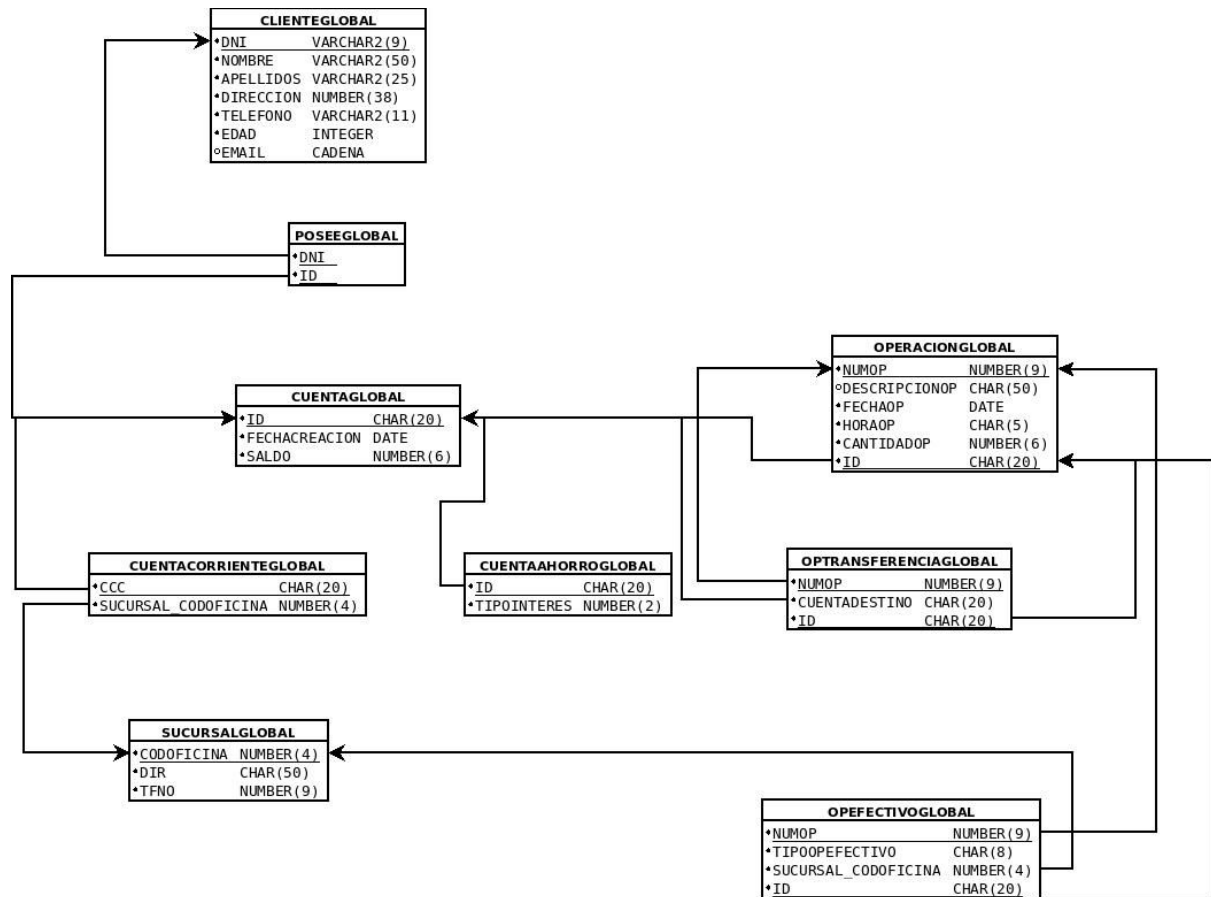
También se debe comentar que las especializaciones no son consistentes porque por ejemplo en la tabla cuenta, no se comprueba que se clasifiquen en corriente o ahorro o pueda ser de los dos tipos, de manera similar pasa con la tabla Operación.

5. PARTE 1 - DEFINICIÓN E IMPLEMENTACIÓN DEL ESQUEMA GLOBAL EN ORACLE

El esquema entidad relación global se ha diseñado de manera que se puedan aprovechar todos los atributos y entidades de ambas bases de datos, el resultado es el siguiente:



El esquema relacional global que se ha diseñado es el siguiente:



Uno de los problemas para integrar ambas bases ha sido que en sus respectivas tablas de clientes, una almacena la edad y otra la fecha de nacimiento, por lo que la que almacena la fecha de nacimiento se ha convertido a la edad actual. En esta misma tabla una almacena los dos apellidos por separado y otra solo un apellido, por lo que en la base de datos final salen los apellidos conjuntos tengan uno o los dos.

Otro problema existe es que las tablas utilizan distinto tipo de datos para representar lo mismo, principalmente aquellas que son números. De esta manera, los campos que son números y aparecen conflictos se han transformado a varchar.

Las respectivas tablas de cuentas, utilizan como identificador IBAN y CCC, por lo que en el esquema global se ha decidido que ambas se combinan para formar un único identificador de cuenta, aunque es una decisión complicada.

En OpEfectivo, el tipo de la base de datos de schema2bd2 no se sabe el caso de retirada porque no hay ningún dato insertado, sin embargo, se sabe que el de ingreso es 'ingreso'.

El código SQL es el siguiente:

```

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW OpTransferenciaGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;

```

```
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW OpEfectivoGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW OperacionGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW ClienteGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW CuentaCorrienteGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW CuentaAhorroGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW CuentaGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/

BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW PoseeGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/
```

```
BEGIN
    EXECUTE IMMEDIATE 'DROP VIEW SucursalGlobal';
EXCEPTION
    WHEN OTHERS THEN NULL;
END;
/
```

```
CREATE VIEW ClienteGlobal AS
SELECT T.dni, T.nombre, T.apellido1 || ' ' || T.apellido2 as apellidos,
D.calle || ' ' || D.numero || ' ' || D.piso || ' ' || D.ciudad as direccion, T.telefono,
(months_between(TRUNC(sysdate), to_date(T.fecha_nacimiento,'DD-MON-YYYY'))/12 - 1900) as
edad, NULL as email
FROM titular@schema2bd2 T, direccion@schema2bd2 D
WHERE T.direccion = D.id_direccion
UNION
SELECT dni, nombre, apellido as apellidos, direccion, telefono, edad, email
FROM Cliente;
```

```
CREATE VIEW CuentaGlobal AS
SELECT C.ccc as id, C.fechacreacion, C.saldo
FROM cuenta@schema2bd2 C
UNION
SELECT iban as id, fechacreacion, saldo
FROM cuenta;
```

```
CREATE VIEW PoseeGlobal AS
SELECT ccc as id, titular as dni
FROM cuenta@schema2bd2
UNION
SELECT iban as id, dni
FROM Posee;
```

```
CREATE VIEW CuentaCorrienteGlobal AS
SELECT C.ccc as id, cast(C.sucursal_codoficina as varchar(15)) as sucursal_codoficina
FROM cuentacorriente@schema2bd2 C
UNION
SELECT iban as id, oficina as sucursal_codoficina
FROM cuenta
WHERE tipo = 'corriente';
```

```
CREATE VIEW CuentaAhorroGlobal AS
SELECT C.ccc as id, C.tipointeres
FROM cuentaahorro@schema2bd2 C
UNION
SELECT iban as id, interes as tipointeres
FROM cuenta
```

```
WHERE tipo = 'ahorro';
```

```
CREATE VIEW SucursalGlobal AS
SELECT cast(S.codoficina as varchar(15)) as codoficina, S.dir, cast(S.tfn as varchar(15))
as tfno
FROM sucursal@schema2bd2 S
UNION
SELECT codigo as codoficina, direccion as dir, telefono as tfno
FROM oficina;
```

```
CREATE VIEW OperacionGlobal AS
SELECT cast(numop as varchar(100)) as numop, descripcionop, fechaop, horaop, cantidadop,
ccc as id
FROM operacion@schema2bd2
UNION
SELECT codigo as numop, descripcion as descripcionop, fecha as fechaop, hora as horaop,
cantidad as cantidadop, IBANOrigen as id
FROM Operacion;
```

```
CREATE VIEW OpTransferenciaGlobal AS
SELECT cast(numop as varchar(100)) as numop, cuentadestino, ccc as id
FROM optransferencia@schema2bd2
UNION
SELECT codigo as numop, IBANDestino as cuentadestino, IBANOrigen as id
FROM Operacion;
```

```
CREATE VIEW OpEfectivoGlobal AS
SELECT cast(numop as varchar(100)) as numop, tipoopefectivo, cast(sucursal_codoficina as
varchar(10)) as sucursal_codoficina, ccc as id
FROM opefectivo@schema2bd2
UNION
SELECT codigo as numop, tipo as tipoopefectivo, oficina as sucursal_codoficina, IBANOrigen as
id
FROM Operacion
WHERE tipo = 'ingreso' or tipo = 'retirada';
```

- PRUEBAS

Con la consulta:

```
SELECT *
FROM PoseeGlobal
WHERE ROWNUM <=20;
```

```

ID
-----
DNI
-----
876333338600000014
34535111B

AD11 0391 5216 RDAZ WGJK QBJH
88-0493337

AD35 2063 5569 HWAY OAQW YX1D
06-4451712

```

Se observa la correcta integración de ambas bases de datos.

La tabla Cliente:

*SELECT **
FROM ClienteGlobal;

<u>DNI</u>	<u>NOMBRE</u>	<u>APELLIDOS</u>	<u>DIRECCION</u>	<u>TELEFONO</u>	<u>EDAD</u>	<u>EMAIL</u>
12-8819012	Ashby	Wittlinger	5 Roxbury Pass	442-289-1099	65	
12-9463236	Sarina	Purcer	1109 Veith Crossing	725-575-1073	44	spurcer22@marketwatch.com
12341111B	Juan	Ferta Podre	Paseo de las Olivas 23 4D Zaragoza	643123456	50.37	
12381111B	Ana	Castrolidi Rupini	Paseo de las Olivas 28 8B Zaragoza	643123456	42.7	
13-4399287	Charin	Barensky	26 Lakewood Gardens Terrace	389-913-7270	63	charenskyoc@soundcloud.com
13-5086325	Heidi	Rosenauer	84 Spenser Drive	142-320-1002	47	
13-6133403	Tami	Crawshaw	04 Oak Valley Road	111-269-0976	67	tcrawshaw8u@patch.com
13411111B	Manuel	Sua Mustio	Avenida Balmes 34 sexto C Madrid	643123456	57.8	
13533111B	Juan	Rico Afortunado	Paseo de las Olivas 34 5D Zaragoza	643123456	55.29	

Se puede observar que las direcciones han sido concatenadas y la edad ha sido calculada. Los teléfonos son de diferente formato pero ambos son varchar.

Finalmente, se ha probado la tabla OpEfectivoGlobal:

*SELECT **
FROM OpEfectivoGlobal;

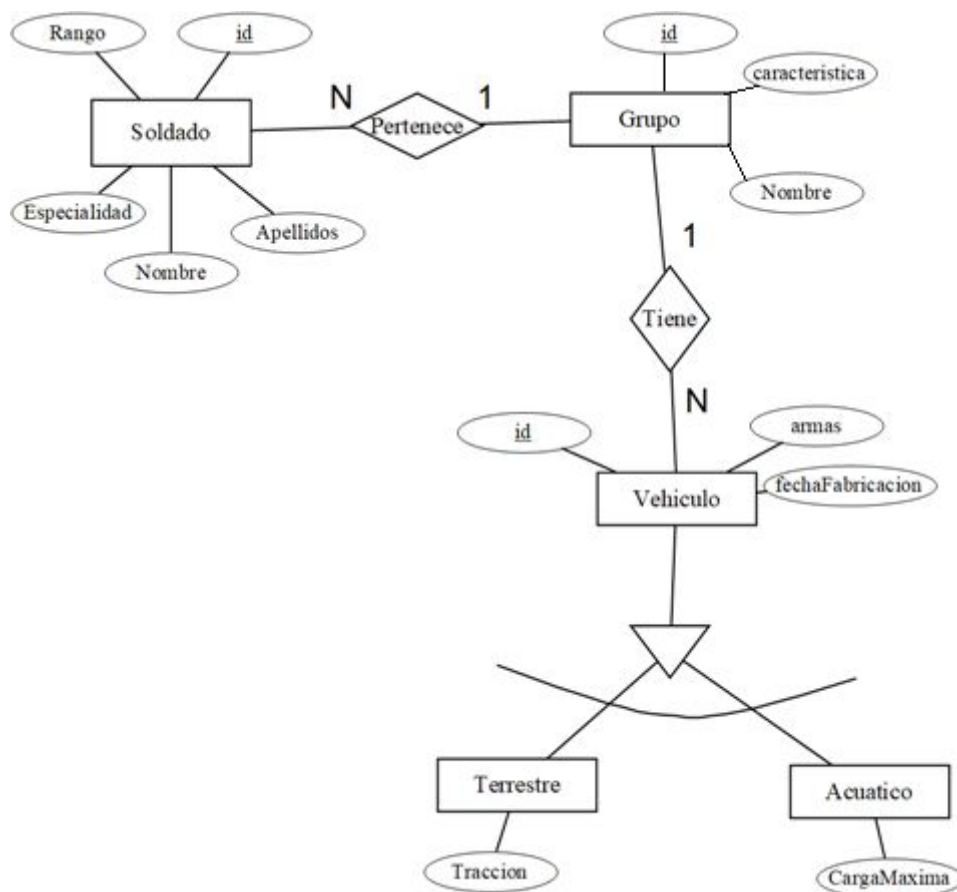
<u>NUMOP</u>	<u>TIPOOPEFECTIVO</u>	<u>SUCURSAL</u>	<u>CODOFI</u>	<u>ID</u>
1	ingreso	8063	GR20 4480 188V	SQSO VCUB ITWH JOX
1	ingreso	1111		8,763111135E+017
1	ingreso	2222		8,763111131E+017
10	ingreso	1821	DK42 0014 8141 2472	94
100	retirada	4079	ES64 2033 2865 1578 2727	2945
101	ingreso	1821	SM07 Y124 8166 683P OH9H XNNC	ZB4
103	retirada	9240	AL03 2690 6270 8YR8 IH05 IEVK	I52U
105	retirada	2605	CY47 6987 8751 VVLO EF9Z VEDU	ZOLA

Se puede comprobar que se han combinado de manera correcta ambas bases.

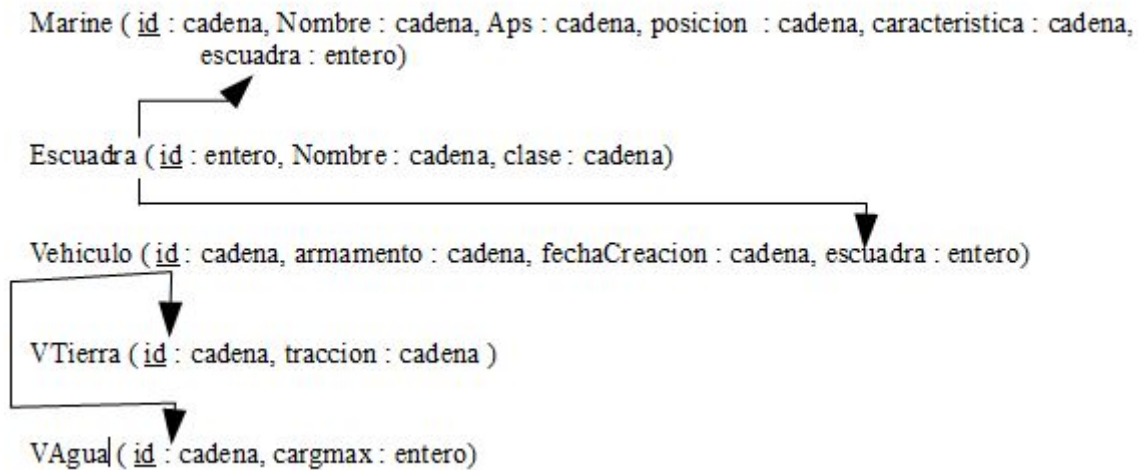
6. PARTE 2 - ENUNCIADO DE UN PROBLEMA DE DISEÑO DE BASE DE DATOS

El ministerio de defensa quiere almacenar los datos de sus soldados, grupos y vehículos en una nueva base de datos más ligera para que las consultas tengan mayor rapidez. Así, quiere una base de datos formada por soldados (los cuales tienen un identificador, un rango, una especialidad (tirador, explosivos, etc) y por supuesto su nombre y apellido) los cuales pertenecen a un grupo que tiene un identificador, una característica determinada (granaderos, apoyo, etc.) y un nombre asociado. A su vez, los grupos pueden tener vehículos asociados. Estos vehículos tienen un identificador (podría ser la matrícula), unas armas montadas en ellos (si no tiene armas tiene que poner NO) y una fecha de fabricación. Además, estos vehículos pueden ser de tierra o agua. Si son de tierra se debe guardar su tipo de tracción y si son de agua se debe guardar la máxima carga que deben soportar.

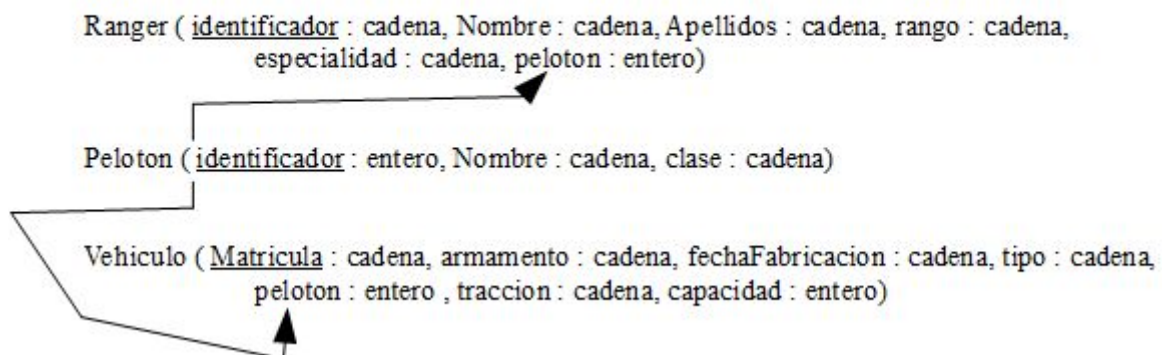
7. PARTE 2 - ESQUEMA CONCEPTUAL PARA EL PROBLEMA ENUNCIADO



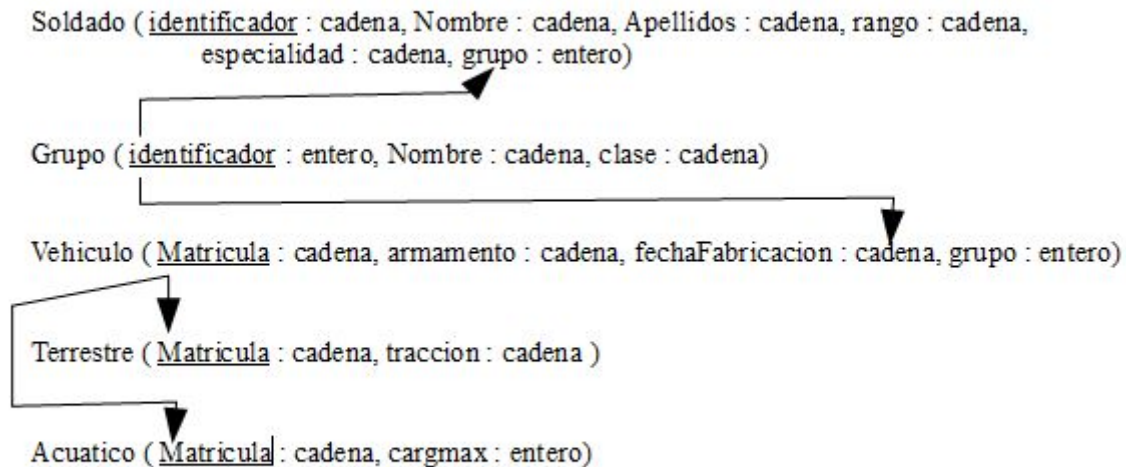
8. PARTE 2 - ESQUEMA LÓGICO 1 PARA EL PROBLEMA ENUNCIADO



9. PARTE 2 - ESQUEMA LÓGICO 2 PARA EL PROBLEMA ENUNCIADO



10. PARTE 2 - ESQUEMA GLOBAL Y SU IMPLEMENTACIÓN EN POSTGRESQL



```
CREATE VIEW Grupo AS
SELECT *
    FROM dblink('dbname=Marina',
                'select id, Nombre, caracteristica from Escuadra')
    AS t1(id integer, Nombre varchar(100), caracteristica varchar(100))
UNION
SELECT *
    FROM dblink('dbname=Rangers',
                'select identificador, nombre, clase from Peloton')
    AS t2(identificador integer, nombre varchar(100), clase varchar(100));

CREATE VIEW Soldado AS
SELECT *
    FROM dblink('dbname=Rangers',
                'select identificador, nombre, apellidos, rango, especialidad, peloton from
Ranger')
    AS t1(identificador integer, nombre varchar(100), apellidos varchar(100), rango
varchar(50), especialidad varchar(100), grupo integer)
UNION
SELECT *
    FROM dblink('dbname=Marina',
                'select id, Nombre, Aps, posicion, caracteristica, escuadra from Marine')
    AS t2(id integer, Nombre varchar(100), Aps varchar(100), posicion varchar(50),
caracteristica varchar(100), escuadra integer);

CREATE VIEW Vehiculo AS
SELECT *
    FROM dblink('dbname=Rangers',
                'select Matricula, armamento, fechaFabricacion, peloton from Vehiculo')
    AS t1(Matricula integer, armamento varchar(100), fechaFabricacion date, grupo integer)
```

```
UNION
SELECT *
  FROM dblink('dbname=Marina',
              'select id, armas, fechaCreacion, escuadra from Vehiculo')
  AS t2(id integer, armas varchar(100), fechaCreacion date, escuadra integer);

CREATE VIEW Terrestre AS
SELECT *
  FROM dblink('dbname=Rangers',
              'select Matricula, traccion from Vehiculo where tipo='Tierra'')
  AS t1(Matricula integer, traccion varchar(100))
UNION
SELECT *
  FROM dblink('dbname=Marina',
              'select id, traccion from VTierra')
  AS t2(id integer, traccion varchar(100));

CREATE VIEW Acuatico AS
SELECT *
  FROM dblink('dbname=Rangers',
              'select Matricula, capacidad from Vehiculo where tipo='Agua'')
  AS t1(Matricula integer, cargamax varchar(100))
UNION
SELECT *
  FROM dblink('dbname=Marina',
              'select id, cargamax from VAgua')
  AS t2(id integer, cargamax varchar(100));
```

11. ACTUALIZACIONES DE DATOS SOBRE EL ESQUEMA GLOBAL

- ORACLE

Al insertar datos sobre el esquema global, se pueden llegar a dar problemas de consistencia, ya que por ejemplo, si en la tabla Cliente se inserta un usuario, se puede dar que se inserten en ambas bases federadas en una con el email y en la otra sin el email, o por ejemplo, la fecha de nacimiento se debería insertar en la base de datos Banquete ya que la otra soporta únicamente la edad. Se va a visualizar con el siguiente ejemplo:

Se tiene el siguiente cliente, traducido a su insert sobre el esquema global:

```
insert into ClienteGlobal(dni, nombre, apellidos, direccion, telefono, edad, email) VALUES
('123456789A', 'Pepito', 'Popoto Pepete', 'c/Falsa 123 Sevilla', '123456789', 25,
'pepito@servidor.com');
```

En este caso, como los clientes poseen todos los campos del esquema local de banquito, se puede introducir una única tupla en esta base y ver el resultado en el esquema global:

```
insert into Cliente(dni, nombre, apellido, direccion, telefono, edad, email) VALUES  
('123456789A', 'Pepito', 'Popoto Pepete', 'c/Falsa 123 Sevilla', '123456789', 25,  
'pepito@servidor.com');
```

Y se comprueba que se ha actualizado la vista:

```
select * from ClienteGlobal where nombre='Pepito';
```

Otro problema de insertar directamente sobre el esquema global es que se debe comprobar que la clave primaria está en alguna de las bases de datos federadas, si no, puede dar lugar a consistencias de que tengan la misma clave primaria y atributos completamente distintos, por lo que no se sabe con que tupla trabajar.

En el esquema global, las cuentas poseen un identificador, por lo que el valor semántico de IBAN y CCC se pierde. Si se desea insertar una nueva cuenta daría igual en cual de las dos bases se realice, ya que no tiene implicación semántica.

- POSTGRESQL

Tabla grupo de Schema2DB2

id	nombre	caracteristica
1004	Cowboy 1	Artificieros
1005	Cowboy 2	Artilleros
1007	Alpha 1	Tiradores
4	Aguila 4	Fusileros
1	Aguila 1	Artificieros
2	Aguila 2	Artilleros
1006	Bravo 1	Tiradores
3	Aguila 3	Apoyo

(8 filas)

Actualizacion de Marine en Marina

```
UPDATE Escuadra SET Nombre='Oso 3' WHERE id='1';
```

id	nombre	caracteristica
1	Aguila 1	Artificieros
2	Aguila 2	Artilleros
3	Aguila 3	Apoyo
4	Aguila 4	Fusileros

id	nombre	caracteristica
----	--------	----------------

2	Aguila 2	Artilleros
3	Aguila 3	Apoyo
4	Aguila 4	Fusileros
1	Oso 3	Artificieros

Actualizacion de Peloton en Rangers

UPDATE Peloton SET Clase='Fusileros' WHERE identificador='1004';

identificador	nombre	clase
1004	Cowboy 1	Artificieros
1005	Cowboy 2	Artilleros
1006	Bravo 1	Tiradores
1007	Alpha 1	Tiradores

identificador	nombre	clase
1005	Cowboy 2	Artilleros
1006	Bravo 1	Tiradores
1007	Alpha 1	Tiradores
1004	Cowboy 1	Fusileros

Tabla Grupo Actualizada en Schema2DB2

id	nombre	caracteristica
1005	Cowboy 2	Artilleros
1	Oso 3	Artificieros
1007	Alpha 1	Tiradores
4	Aguila 4	Fusileros
1004	Cowboy 1	Fusileros
2	Aguila 2	Artilleros
1006	Bravo 1	Tiradores
3	Aguila 3	Apoyo

Tabla soldado de Schema2DB2

identificador	nombre	apellidos	rango	especialidad	grupo
7956	Mario	San Francisco	Soldado raso	Ametralladora	1005
5910689	Javier	Lopez	Sargento Primero	Tirador	2
94653	Jesus	Nazaret	Sargento Primero	Tirador	1005

5976438	Juan	Gonzalez	Sargento	Tirador	2
104575	Antonio	Ferrer	Sargento	Conductor	1005
73594	Jose	Fernandez	Cabo	Tirador	1005
27653491	Esteban	Coesta cupado	Cabo	Conductor	2
15467	Daniel	Ramos	Cabo	Granadero	1005
16435957	Marcos	Garcia	Cabo	Granadero	2
13452312	Francisco	Perez	Sargento	Tirador	2

Tabla marine de Marina

UPDATE Marine SET Nombre='Juan Jose' WHERE id='05910689';

id	nombre	aps	posicion	caracteristica	escuadra
13452312	Francisco	Perez	Sargento	Tirador	2
05910689	Javier	Lopez	Sargento Primero	Tirador	2
16435957	Marcos	Garcia	Cabo	Granadero	2
05976438	Juan	Gonzalez	Sargento	Tirador	2
27653491	Esteban	Coesta cupado	Cabo	Conductor	2

id	nombre	aps	posicion	caracteristica	escuadra
13452312	Francisco	Perez	Sargento	Tirador	2
16435957	Marcos	Garcia	Cabo	Granadero	2
05976438	Juan	Gonzalez	Sargento	Tirador	2
27653491	Esteban	Coesta cupado	Cabo	Conductor	2
05910689	Juan Jose	Lopez	Sargento Primero	Tirador	2

Tabla marine de Marina

UPDATE Ranger SET rango='cabo' WHERE identificador='7956';

identificador	nombre	apellidos	rango	especialidad	peloton
7956	Mario	San Francisco	Soldado raso	Ametralladora	1005
15467	Daniel	Ramos	Cabo	Granadero	1005
73594	Jose	Fernandez	Cabo	Tirador	1005
94653	Jesus	Nazaret	Sargento Primero	Tirador	1005
104575	Antonio	Ferrer	Sargento	Conductor	1005

identificador	nombre	apellidos	rango	especialidad	peloton
15467	Daniel	Ramos	Cabo	Granadero	1005
73594	Jose	Fernandez	Cabo	Tirador	1005
94653	Jesus	Nazaret	Sargento Primero	Tirador	1005

104575	Antonio	Ferrer	Sargento	Conductor	1005
7956	Mario	San Francisco	cabo	Ametralladora	1005

Tabla soldado actualizada de Schema2DB2

identificador	nombre	apellidos	rango	especialidad	grupo
94653	Jesus	Nazaret	Sargento Primero	Tirador	1005
5976438	Juan	Gonzalez	Sargento	Tirador	2
104575	Antonio	Ferrer	Sargento	Conductor	1005
73594	Jose	Fernandez	Cabo	Tirador	1005
7956	Mario	San Francisco	cabo	Ametralladora	1005
27653491	Esteban	Coesta cupado	Cabo	Conductor	2
15467	Daniel	Ramos	Cabo	Granadero	1005
16435957	Marcos	Garcia	Cabo	Granadero	2
13452312	Francisco	Perez	Sargento	Tirador	2
5910689	Juan Jose	Lopez	Sargento Primero	Tirador	2

Tabla vehiculo de Schema2DB2

matricula	armamento	fechafabricacion	grupo
1345	Lanzacohetes	2000-01-03	1005
14054	lanzagranadas	2007-02-18	2
14055	ametralladora	2007-02-18	2
1234	Ametralladora Superior	2010-03-01	1005
1579	NO	2018-01-21	1005
14057	ametralladora	2007-02-18	2
6741	Ametralladora	2012-05-30	1005
6479	Ametralladora Superior	2013-12-01	1006
14056	lanzagranadas	2007-02-18	2

Tabla vehiculo de Marina

UPDATE Vehiculo SET armas='NO' WHERE id='14057';

id	armas	fechacreacion	escuadra
14054	lanzagranadas	2007-02-18	2
14055	ametralladora	2007-02-18	2
14056	lanzagranadas	2007-02-18	2
14057	ametralladora	2007-02-18	2

id	armas	fechacreacion	escuadra
14054	lanzagranadas	2007-02-18	2

```
14055 | ametralladora | 2007-02-18 | 2
14056 | lanzagranadas | 2007-02-18 | 2
14057 | NO | 2007-02-18 | 2
```

Tabla vehiculo de Rangers

UPDATE Vehiculo SET armamento='NO' WHERE Matricula='1345';

```
matricula | armamento | fechafabricacion | tipo | peloton | traccion | capacidad
-----+-----+-----+-----+-----+-----+-----
1234 | Ametralladora Superior | 2010-03-01 | Tierra | 1005 | 4 Ruedas |
1345 | Lanzacohetes | 2000-01-03 | Tierra | 1005 | Delantera |
6479 | Ametralladora Superior | 2013-12-01 | Tierra | 1006 | 4 Ruedas |
1579 | NO | 2018-01-21 | Agua | 1005 | | 1000
6741 | Ametralladora | 2012-05-30 | Agua | 1005 | | 3000
```

```
matricula | armamento | fechafabricacion | tipo | peloton | traccion | capacidad
-----+-----+-----+-----+-----+-----+-----
1234 | Ametralladora Superior | 2010-03-01 | Tierra | 1005 | 4 Ruedas |
6479 | Ametralladora Superior | 2013-12-01 | Tierra | 1006 | 4 Ruedas |
1579 | NO | 2018-01-21 | Agua | 1005 | | 1000
6741 | Ametralladora | 2012-05-30 | Agua | 1005 | | 3000
1345 | NO | 2000-01-03 | Tierra | 1005 | Delantera |
```

Tabla vehiculo actualizada de Schema2DB2

```
matricula | armamento | fechafabricacion | grupo
-----+-----+-----+-----
14054 | lanzagranadas | 2007-02-18 | 2
14055 | ametralladora | 2007-02-18 | 2
1234 | Ametralladora Superior | 2010-03-01 | 1005
1579 | NO | 2018-01-21 | 1005
1345 | NO | 2000-01-03 | 1005
14057 | NO | 2007-02-18 | 2
6741 | Ametralladora | 2012-05-30 | 1005
6479 | Ametralladora Superior | 2013-12-01 | 1006
14056 | lanzagranadas | 2007-02-18 | 2
```

En el caso de insertar en el esquema global directamente, aparece el problema de que el soldado, grupo o vehículo es una entidad muy generalizada y no llega a pertenecer ni a la Marina ni a los Rangers, que son los dos grupos subyacentes.

12. DIFICULTADES ENCONTRADAS

Una de las principales dificultades ha sido realizar la conversión de tipos en Oracle, ya que al ser incompatibles, para crear el esquema global deben ser del mismo tipo. También ha costado una ligera parte del tiempo deducir el esquema local de oracle a través de las consultas, ya que al haber partes mal diseñadas, confunden.

13. ESFUERZOS INVERTIDOS

	Carlos Marañes (717788)	Nicolás Lera (721808)	Total Horas
Parte 1	10h	2h	12h
Parte 2	2h	8h	10h
Redacción	2h	2h	4h
TOTAL horas	14h	12h	26h

14. BIBLIOGRAFÍA

Instalar dblink en PostgreSQL - Último acceso 13/04/2018

<https://stackoverflow.com/questions/3862648/how-to-use-install-dblink-in-postgresql>