

consulta1_tbl.sql

```

-- SET AUTOTRACE ON
SET TRIMSPPOOL ON
SET TRIMOUT ON
SET WRAP OFF

-- servicios usados por cada ciudadano
SELECT DNI_titular, tipoServ
FROM Tarjeta T, Accesos A, Servicio S
WHERE T.idTarjeta = A.idTarjeta AND A.idServ = S.idServ;

-- crear una tabla con los tipos de servicio usados por cada ciudadano (sólo para pruebas: mejor VISTA !!)
CREATE TABLE UsoServCiud AS (
  SELECT DISTINCT DNI_titular, tipoServ
  FROM Tarjeta T, Accesos A, Servicio S
  WHERE T.idTarjeta = A.idTarjeta AND A.idServ = S.idServ
);
SELECT * FROM UsoServCiud;

/*
##### (C1.1)
Parejas (sin repetir) de ciudadanos que han usado exactamente los mismos tipos de servicio.
#####*/
SELECT DISTINCT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2
FROM UsoServCiud C1, UsoServCiud C2
WHERE C1.DNI_titular < C2.DNI_titular AND
  NOT EXISTS((SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C1.DNI_titular
  MINUS
  SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C2.DNI_titular)
  UNION
  (SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C2.DNI_titular
  MINUS
  SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C1.DNI_titular));

/*
##### (C1.2)
#####*/
SELECT DISTINCT C1.DNI_titular, C2.DNI_titular
FROM UsoServCiud C1, UsoServCiud C2
WHERE C1.DNI_titular < C2.DNI_titular AND
  NOT EXISTS((SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular = C1.DNI_titular OR DNI_titular = C2.DNI_titular
  MINUS
  SELECT S1.tipoServ FROM UsoServCiud S1, UsoServCiud S2
  WHERE S1.DNI_titular = C1.DNI_titular AND S2.DNI_titular = C2.DNI_titular
  AND S1.tipoServ = S2.tipoServ));

/*
##### (C1.3)
#####*/
SELECT DISTINCT C1.DNI_titular, C2.DNI_titular
FROM UsoServCiud C1, UsoServCiud C2
WHERE C1.DNI_titular < C2.DNI_titular AND
  NOT EXISTS (SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular = C1.DNI_titular OR DNI_titular = C2.DNI_titular
  MINUS(
  SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C1.DNI_titular
  INTERSECT
  SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = C2.DNI_titular));

/*
##### (C1.4)
#####*/
SELECT DISTINCT C1.DNI_titular, C2.DNI_titular FROM UsoServCiud C1, UsoServCiud C2
WHERE C1.DNI_titular < C2.DNI_titular
MINUS
SELECT C1.DNI_titular, C2.DNI_titular FROM UsoServCiud C1, UsoServCiud C2
WHERE EXISTS (SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular = C1.DNI_titular
  AND tipoServ NOT IN (SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular =C2.DNI_titular))
OR EXISTS (SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular = C2.DNI_titular
  AND tipoServ NOT IN (SELECT tipoServ FROM UsoServCiud
  WHERE DNI_titular =C1.DNI_titular));

```

```

/*                                                                 (C1.5)
#####*/
SELECT DISTINCT C1.DNI_titular, C2.DNI_titular
FROM UsoServCiud C1, UsoServCiud C2 WHERE C1.DNI_titular < C2.DNI_titular
MINUS
SELECT S1.DNI_titular, S2.DNI_titular FROM UsoServCiud S1, UsoServCiud S2
WHERE S1.tipoServ NOT IN (SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = S2.DNI_titular)
OR S2.tipoServ NOT IN (SELECT tipoServ FROM UsoServCiud WHERE DNI_titular = S1.DNI_titular);

/*                                                                 (C1.6)
#####*/
/* crear una vista con las parejas que comparten el uso de algún tipo de servicio, junto con
el número de tipos de servicio que comparten */
CREATE VIEW nServAmbos (DNI_titular1, DNI_titular2, N) AS
SELECT S1.DNI_titular, S2.DNI_titular, count(*) FROM UsoServCiud S1, UsoServCiud S2
WHERE S1.DNI_titular < S2.DNI_titular AND S1.tipoServ = S2.tipoServ
GROUP BY S1.DNI_titular, S2.DNI_titular;
SELECT * FROM nServAmbos;

/* crear una vista con el número de tipos de servicio que ha usado cada ciudadano */
CREATE VIEW nServCiud (DNI_titular, N) AS
SELECT DNI_titular, count(*) FROM UsoServCiud
GROUP BY DNI_titular;
SELECT * FROM nServCiud;

SELECT DNI_titular1, DNI_titular2 FROM nServAmbos, nServCiud nSrv1, nServCiud nSrv2
WHERE DNI_titular1=nSrv1.DNI_titular AND DNI_titular2=nSrv2.DNI_titular
AND nServAmbos.N=nSrv1.N AND nServAmbos.N=nSrv2.N;

DROP VIEW nServAmbos;
DROP VIEW nServCiud;

/*                                                                 (C1.7)
#####*/
/* sin utilizar las vistas anteriores */
SELECT S1.DNI_titular, S2.DNI_titular FROM UsoServCiud S1, UsoServCiud S2
WHERE S1.DNI_titular < S2.DNI_titular AND S1.tipoServ = S2.tipoServ
GROUP BY S1.DNI_titular, S2.DNI_titular
HAVING count(*) = (SELECT count(*) FROM UsoServCiud WHERE DNI_titular = S1.DNI_titular)
AND count(*) = (SELECT count(*) FROM UsoServCiud WHERE DNI_titular = S2.DNI_titular);

/*                                                                 (c1.8)
#####*/
SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2
FROM UsoServCiud c1, UsoServCiud c2
WHERE C1.DNI_titular < C2.DNI_titular
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
FROM UsoServCiud c1, UsoServCiud c2
MINUS
SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
FROM UsoServCiud c1, UsoServCiud c2)
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
FROM UsoServCiud c1, UsoServCiud c2
MINUS
SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
FROM UsoServCiud c1, UsoServCiud c2);

```

```

/*                                                                                                     (c1.9)
#####*/

SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2
FROM UsoServCiud c1, UsoServCiud c2
WHERE C1.DNI_titular < C2.DNI_titular
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      MINUS
      SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular)
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      MINUS
      SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular);

/*                                                                                                     (c1.10)
#####*/
/* crear una vista con las parejas que comparten el uso de algùn tipo de servicio, junto con
   el tipo de servicio que comparten */

CREATE VIEW ServAmbos (DNI_titular1, DNI_titular2, tipoServ) AS
SELECT S1.DNI_titular, S2.DNI_titular, S1.tipoServ FROM UsoServCiud S1, UsoServCiud S2
WHERE S1.DNI_titular < S2.DNI_titular AND S1.tipoServ = S2.tipoServ;

SELECT * FROM ServAmbos;

SELECT DNI_titular1, DNI_titular2
FROM ServAmbos
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      MINUS
      SELECT DNI_titular1, DNI_titular2, tipoServ FROM ServAmbos)
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      MINUS
      SELECT DNI_titular1, DNI_titular2, tipoServ FROM ServAmbos);

/*                                                                                                     (c1.11)
#####*/

SELECT DNI_titular1, DNI_titular2 FROM ServAmbos
MINUS
SELECT DNI_1, DNI_2
FROM (SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C1.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      UNION
      SELECT C1.DNI_titular DNI_1, C2.DNI_titular DNI_2, C2.tipoServ
      FROM UsoServCiud c1, UsoServCiud c2 WHERE C1.DNI_titular < C2.DNI_titular
      MINUS
      SELECT DNI_titular1, DNI_titular2, tipoServ FROM ServAmbos);

/*=====*/
/* deshacer los cambios realizados */
-- eliminar la vista creada
DROP VIEW ServAmbos;
-- eliminar la tabla creada
DROP TABLE UsoServCiud;

```

consulta2.sql

```

-- SET AUTOTRACE ON
SET TRIMSPPOOL ON
SET TRIMOUT ON
SET WRAP OFF
column balanceT      format 990.99

/*
#####
tarjetas de cada contrato en las que la diferencia entre el dinero recargado con la tarjeta
y el gastado con la tarjeta es mayor
#####*/
(C2.1)
SELECT T.idTarjeta, balanceT
FROM tarjeta T
WHERE balanceT = (SELECT max(balanceT) FROM tarjeta WHERE idcontrato = T.idcontrato)
ORDER BY T.idTarjeta;

/*
#####*/
(C2.2)
SELECT idTarjeta, balanceT
FROM (SELECT idTarjeta, balanceT, max(balanceT) OVER (PARTITION BY idcontrato) maxDif
      FROM tarjeta)
WHERE balanceT = maxDif
ORDER BY idTarjeta;

CREATE OR REPLACE TRIGGER nuevaTarjeta
BEFORE INSERT ON tarjeta
FOR EACH ROW
DECLARE tipoC char;
BEGIN
  SELECT tipoContrato INTO tipoC FROM contrato WHERE idContrato = :new.idContrato;
  IF tipoC = 'P' THEN :new.balanceT := 0;
  ELSE :new.balanceT := NULL;
  END IF;
  :new.idTarjeta := :new.idContrato * 10 + :new.numTarj;
END;
/
SHOW ERRORS TRIGGER nuevaTarjeta

CREATE OR REPLACE TRIGGER accesoServicio
AFTER INSERT ON accesos
FOR EACH ROW
DECLARE disponible real;
      elContrato number(5);
      tipoC char;
BEGIN
  elContrato := round(:new.idTarjeta / 10);
  SELECT tipoContrato INTO tipoC FROM contrato WHERE idContrato = elContrato;
  IF tipoC = 'P' THEN
    SELECT saldo INTO disponible FROM contrato WHERE idContrato = elContrato;
    IF disponible + :new.importe > 0 THEN
      UPDATE contrato SET saldo = saldo + :new.importe WHERE idContrato = elContrato;
      UPDATE tarjeta SET balanceT = balanceT + :new.importe WHERE idTarjeta = :new.idTarjeta;
    ELSE
      raise_application_error( -20001, 'no hay saldo disponible');
    END IF;
  ELSE
    raise_application_error( -20002, 'acceso con tarjeta asociada a cuenta no implementado');
  END IF;
END;
/
SHOW ERRORS TRIGGER accesoServicio

```

BD_tarjetas_createBD.sql

```
/* @BD_tarjetas_createBD.sql */
SET ECHO ON
SET LINESIZE 132
SET PAGESIZE 200

/* Eliminar las tablas y vistas de la BD, si existen
*/
@BD_tarjetas_dropBD.sql;

/* Crear las tablas y vistas de la BD
*/
CREATE TABLE BANCO (
  idBanco varchar(24) PRIMARY KEY,
  nombreB varchar(24) NOT NULL,
  URL varchar(64) NOT NULL
);

CREATE TABLE CUENTA (
  IBAN varchar(24) PRIMARY KEY,
  idBanco varchar(24) NOT NULL REFERENCES BANCO(idBanco)
);

CREATE TABLE CONTRATO (
  idContrato number(5) PRIMARY KEY,
  tipoContrato char NOT NULL,
  saldo real,
  IBAN varchar(24) REFERENCES CUENTA(IBAN)
);

CREATE TABLE TITULAR (
  DNI varchar(24) PRIMARY KEY,
  ApNombre varchar(24) NOT NULL,
  fNacim date NOT NULL
);

CREATE TABLE TARJETA (
  idTarjeta number(6) PRIMARY KEY,
  idContrato number(5) NOT NULL REFERENCES CONTRATO(idContrato),
  numTarj number(1) NOT NULL CHECK(numTarj > 0 AND numTarj <=4),
  PIN number(4) NOT NULL,
  DNI_titular varchar(24) NOT NULL REFERENCES TITULAR(DNI),
  balanceT real,
  UNIQUE (idContrato, numTarj),
  UNIQUE (idContrato, DNI_titular)
);

CREATE TABLE tpServ (
  tipoServ varchar(12) PRIMARY KEY
);

CREATE TABLE SERVICIO (
  idServ number(3) PRIMARY KEY,
  tipoServ varchar(12) NOT NULL REFERENCES tpServ(tipoServ),
  infoServ varchar(64)
);

CREATE TABLE MUSEO (
  idServ number(3) PRIMARY KEY REFERENCES SERVICIO(idServ),
  nombreM varchar(24) UNIQUE NOT NULL,
  direccion varchar(64) NOT NULL,
  tfno number(9) NOT NULL
);

CREATE TABLE ACCESOS (
  idTarjeta number(6) REFERENCES TARJETA(idTarjeta),
  fechaHora date,
  idServ number(3) NOT NULL REFERENCES SERVICIO(idServ),
  importe real NOT NULL,
  PRIMARY KEY(idTarjeta, fechaHora)
);
```

```

CREATE OR REPLACE TRIGGER nuevaTarjeta
BEFORE INSERT ON tarjeta
FOR EACH ROW
DECLARE tipoC char;
BEGIN
    SELECT tipoContrato INTO tipoC FROM contrato WHERE idContrato = :new.idContrato;
    IF tipoC = 'P' THEN :new.balanceT := 0;
        ELSE :new.balanceT := NULL;
    END IF;
    :new.idTarjeta := :new.idContrato * 10 + :new.numTarj;
END;
/
SHOW ERRORS TRIGGER nuevaTarjeta

CREATE OR REPLACE TRIGGER accesoServicio
AFTER INSERT ON accesos
FOR EACH ROW
DECLARE disponible real;
    elContrato number(5);
    tipoC char;
BEGIN
    elContrato := round(:new.idTarjeta / 10);
    SELECT tipoContrato INTO tipoC FROM contrato WHERE idContrato = elContrato;
    IF tipoC = 'P' THEN
        SELECT saldo INTO disponible FROM contrato WHERE idContrato = elContrato;
        IF disponible + :new.importe > 0 THEN
            UPDATE contrato SET saldo = saldo + :new.importe WHERE idContrato = elContrato;
            UPDATE tarjeta SET balanceT = balanceT + :new.importe WHERE idTarjeta = :new.idTarjeta;
        ELSE
            raise_application_error( -20001, 'no hay saldo disponible');
        END IF;
    ELSE
        raise_application_error( -20002, 'acceso con tarjeta asociada a cuenta no implementado');
    END IF;
END;
/
SHOW ERRORS TRIGGER accesoServicio
-- mostrar la información básica de las tablas y vistas
@BD_tarjetas_infoBD.sql

```

BD_tarjetas_infoBD.sql

```

/* @BD_tarjetas_infoBD.sql */
-- SET ECHO ON

-- mostrar la información básica de las tablas y vistas
DESCRIBE MUSEO;
DESCRIBE ACCESOS;
DESCRIBE SERVICIO;
DESCRIBE TITULAR;
DESCRIBE TARJETA;
DESCRIBE CONTRATO;
DESCRIBE CUENTA;
DESCRIBE BANCO;

```

BD_tarjetas_deleteData.sql

```

/* @BD_tarjetas_deleteData.sql */
-- SET ECHO ON

-- eliminar los datos de las tablas en el orden adecuado, para no violar las restricciones
DELETE FROM MUSEO;
DELETE FROM ACCESOS;
DELETE FROM SERVICIO;
DELETE FROM TARJETA;
DELETE FROM TITULAR;
DELETE FROM CONTRATO;
DELETE FROM CUENTA;
DELETE FROM BANCO;

-- mostrar los datos de las tablas y vista
@BD_tarjetas_showData.sql

```

BD_tarjetas_insertData.sql

```

/* @BD_tarjetas_insertData.sql */
SET ECHO ON

/* eliminar los datos existentes en las tablas
*/
-- @BD_tarjetas_deleteData.sql
/* añadir los datos de prueba a las tablas
*/

INSERT INTO tpServ VALUES ('MUSEO');
INSERT INTO tpServ VALUES ('AUTOBUS');
INSERT INTO tpServ VALUES ('TRANVIA');
INSERT INTO tpServ VALUES ('RECARGA');

INSERT INTO BANCO VALUES ('2085', 'Ibercaja', 'www.ibercaja.es');
INSERT INTO BANCO VALUES ('2086', 'CAI', 'www.cai.es');
INSERT INTO BANCO VALUES ('0049', 'Santander', 'www.bancosantander.es');

INSERT INTO CUENTA VALUES ('ES0557632', '2085');
INSERT INTO CUENTA VALUES ('ES0597642', '2086');
INSERT INTO CUENTA VALUES ('ES3295271', '2086');
INSERT INTO CUENTA VALUES ('ES0193281', '0049');

INSERT INTO CONTRATO VALUES (101, 'P',0.00,'');
INSERT INTO CONTRATO VALUES (102, 'C', '', 'ES0557632');
INSERT INTO CONTRATO VALUES (103, 'P',0.00,'');
INSERT INTO CONTRATO VALUES (104, 'P',0.00,'');
INSERT INTO CONTRATO VALUES (105, 'P',0.00,'');
INSERT INTO CONTRATO VALUES (106, 'C', '', 'ES0193281');

INSERT INTO TITULAR VALUES (17458671, 'Pérez, Ángel', TO_DATE('25/08/1954', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (18654231, 'Ruiz, Santiago', TO_DATE('14/03/1987', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (19654368, 'López, Luis', TO_DATE('18/06/1999', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (18421389, 'Ramón, Sara', TO_DATE('23/11/1990', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (23421387, 'Rivas, María', TO_DATE('14/02/2000', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (22657943, 'García, Juan', TO_DATE('25/07/2003', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (23459278, 'García, Juan', TO_DATE('23/04/1979', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (25456873, 'García, Andrés', TO_DATE('07/12/2004', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (19563478, 'Sánchez, María', TO_DATE('12/10/1981', 'dd/mm/yyyy'));
INSERT INTO TITULAR VALUES (21356987, 'Ruiz, Elena', TO_DATE('26/08/2004', 'dd/mm/yyyy'));

INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (101, 1, 1234, 17458671);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (103, 1, 1234, 18654231);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (103, 2, 1234, 21356987);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (102, 1, 1234, 18421389);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (104, 1, 1234, 19563478);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (105, 1, 1234, 23459278);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (105, 2, 1234, 25456873);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (105, 3, 1234, 22657943);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (103, 3, 1234, 19654368);
INSERT INTO TARJETA(idContrato, numTarj, PIN, DNI_titular) VALUES (106, 1, 1234, 23421387);

INSERT INTO SERVICIO VALUES (501, 'MUSEO', 'BELLAS ARTES');
INSERT INTO SERVICIO VALUES (502, 'MUSEO', 'BELLAS ARTES');
INSERT INTO SERVICIO VALUES (503, 'MUSEO', 'BELLAS ARTES');
INSERT INTO SERVICIO VALUES (504, 'MUSEO', 'PALEONTOLÓGICO');
INSERT INTO SERVICIO VALUES (505, 'MUSEO', 'HISTORIA');
INSERT INTO SERVICIO VALUES (506, 'MUSEO', 'TAPICES');
INSERT INTO SERVICIO VALUES (600, 'AUTOBUS', 'linea 23');
INSERT INTO SERVICIO VALUES (601, 'AUTOBUS', 'linea 42');
INSERT INTO SERVICIO VALUES (602, 'AUTOBUS', 'linea 43');
INSERT INTO SERVICIO VALUES (603, 'AUTOBUS', 'linea 44');
INSERT INTO SERVICIO VALUES (650, 'TRANVIA', 'L1');
INSERT INTO SERVICIO VALUES (651, 'TRANVIA', 'L2');
INSERT INTO SERVICIO VALUES (999, 'RECARGA', '');

INSERT INTO MUSEO VALUES (501, 'CAMÓN AZNAR', 'Calle de Espoz y Mina, 23', 976397387);
INSERT INTO MUSEO VALUES (502, 'PABLO GARGALLO', 'Plaza San Felipe, 3', 976724922);
INSERT INTO MUSEO VALUES (503, 'PABLO SERRANO', 'Paseo María Agustín, 20', 976280659);
INSERT INTO MUSEO VALUES (504, 'PALEONTOLÓGICO', 'Pedro Cerbuna 12 (fac. Ciencias), 976762122);
INSERT INTO MUSEO VALUES (505, 'ZARAGOZA', 'Plaza de los Sitios, 6', 976222181);
INSERT INTO MUSEO VALUES (506, 'TAPICES DE LA SEO', 'Plaza de la Seo, s/n.', 976291231);

```

```

INSERT INTO ACCESOS VALUES (1011, TO_DATE('20/06/2014 9:05', 'dd/mm/yyyy HH24:MI'), 999, 10.00);
INSERT INTO ACCESOS VALUES (1011, TO_DATE('20/06/2014 9:20', 'dd/mm/yyyy HH24:MI'), 600, -0.75);
INSERT INTO ACCESOS VALUES (1011, TO_DATE('20/06/2014 9:30', 'dd/mm/yyyy HH24:MI'), 650, 0.00);
INSERT INTO ACCESOS VALUES (1051, TO_DATE('20/06/2014 12:20', 'dd/mm/yyyy HH24:MI'), 999, 30.00);
INSERT INTO ACCESOS VALUES (1052, TO_DATE('20/06/2014 13:15', 'dd/mm/yyyy HH24:MI'), 600, -0.75);
INSERT INTO ACCESOS VALUES (1052, TO_DATE('20/06/2014 13:20', 'dd/mm/yyyy HH24:MI'), 650, 0.00);
INSERT INTO ACCESOS VALUES (1053, TO_DATE('20/06/2014 16:16', 'dd/mm/yyyy HH24:MI'), 600, -0.75);
INSERT INTO ACCESOS VALUES (1053, TO_DATE('20/06/2014 16:40', 'dd/mm/yyyy HH24:MI'), 650, -0.75);
INSERT INTO ACCESOS VALUES (1031, TO_DATE('21/06/2014 11:50', 'dd/mm/yyyy HH24:MI'), 999, 20.00);
INSERT INTO ACCESOS VALUES (1031, TO_DATE('21/06/2014 13:10', 'dd/mm/yyyy HH24:MI'), 600, -0.75);

```

COMMIT;

/* mostrar los datos de las tablas y vista

*/

@BD_tarjetas_showData.sql

BD_tarjetas_showData.sql

/* @BD_tarjetas_showData.sql */

-- SET ECHO ON

SET LINESIZE 132

SET PAGESIZE 200

```

column idBanco      format A12
column nombreB      format a16
column URL           format a40
column saldo         format 990.99
column balanceT     format 990.99
column importe      format 990.99

```

SET WRAP OFF

/* mostrar los datos de las tablas y vistas

*/

SELECT * FROM BANCO;

SELECT * FROM CUENTA;

SELECT * FROM CONTRATO;

SELECT DNI, APNOMBRE, TO_CHAR(FNACIM, 'dd/mm/yyyy') FNACIM FROM titular;

SELECT * FROM TARJETA;

SELECT * FROM SERVICIO;

SELECT * FROM MUSEO;

SELECT idTarjeta,
TO_CHAR(FECHAHORA, 'dd/mm/yyyy hh:mi') "FECHA y HORA", IDSERV, IMPORTE FROM ACCESOS;

BD_tarjetas_dropBD.sql

/* @BD_tarjetas_dropBD.sql */

-- SET ECHO ON

/* eliminar las tablas y vistas creadas, en el orden
adecuado, para no violar las restricciones

*/

DROP TABLE MUSEO;

DROP TABLE ACCESOS;

DROP TABLE SERVICIO;

DROP TABLE TARJETA;

DROP TABLE TITULAR;

DROP TABLE CONTRATO;

DROP TABLE CUENTA;

DROP TABLE BANCO;

DROP TABLE tpServ;

DROP TRIGGER nuevaTarjeta;

DROP TRIGGER accesoServicio;

/* mostrar la información básica de las tablas y vistas

*/

-- @BD_tarjetas_infoBD.sql