

PRÁCTICA 1

APARTADO 1

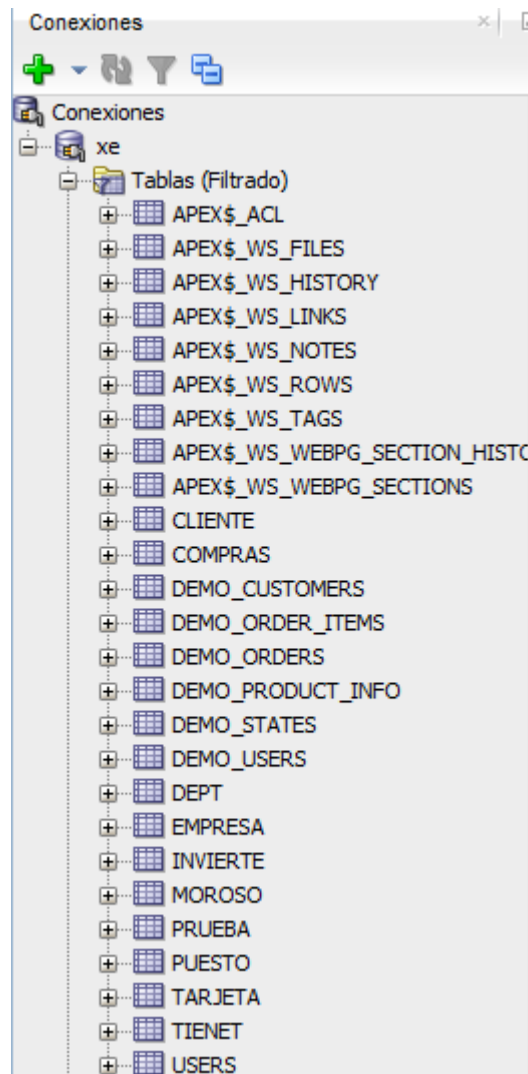
Conectarse a Oracle desde un laboratorio.

APARTADO 2. Reconocer la clave primaria (CP), clave ajena (CA), claves candidatas (CC), dependencia funcional (DF) y dependencia multivalorada (DM), datos erróneos

Utilizando la BDejemplo.sql tal y como está definida, sin CP ni CA, (decide tu cuáles deben ser las CP en los apartados que lo necesites) realiza los siguientes apartados:

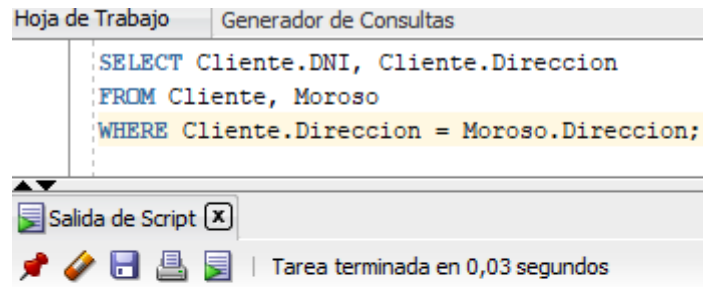
a) Crea la BDejemplo en Oracle.

Ejecutamos el script BDejemplo.sql y comprobamos que se hayan creado todas las tablas.



- b) Describe lo indicado en cada apartado:
b.1) Qué resultado da la siguiente consulta.

```
Select Cliente.DNI, Cliente.Direccion
from Cliente, Moroso
where Cliente.Direccion = Moroso.Direccion;
```



DNI	DIRECCION
00000001	direc 11
00000003	direc 13
00000005	direc 11
00000006	direc 16

El resultado de la consulta devuelve el DNI '0..1' que no está en Moroso: hay unión con pérdida información, por hacer la unión de dos tablas por un atributo que no es CP ni CC en ninguna de las dos tabla: Datos erróneos.

```
REM ... Cliente: CL(DNI, NombreC,Direccion,Telefono)

INSERT INTO Cliente VALUES ('00000001','Client A','direc 11','911111111111');
INSERT INTO Cliente VALUES ('00000003','Client B','direc 13','911111111113');
INSERT INTO Cliente VALUES ('00000002','Client C','direc 12','911111111112');
INSERT INTO Cliente VALUES ('00000005','Client A','direc 11','911111111115');
INSERT INTO Cliente VALUES ('00000004','Client A','direc 14','911111111114');
INSERT INTO Cliente VALUES ('00000006','Client D','direc 16','911111111116');
```

```
REM ... Moroso: MO(DNI, NombreC,Direccion,Telefono)

INSERT INTO Moroso VALUES ('00000003','Client B','direc 13','911111111113');
INSERT INTO Moroso VALUES ('00000007','Client E','direc 17','911111111117');
INSERT INTO Moroso VALUES ('00000005','Client A','direc 11','911111111115');
INSERT INTO Moroso VALUES ('00000006','Client D','direc 16','911111111116');
```

b.2) En la tabla Cliente, ¿Cuándo podemos considerar CC el atributo Dirección? Describe la respuesta con la relación hay entre los valores de los clientes y los valores de las direcciones donde viven.

Cuando el resto de atributos dependan de la dirección, sucede esto Cuando cada cliente viva en una dirección distinta siempre. Cada fila tiene Dirección distinta.

b.3) En la tabla Puesto, cada título tiene el mismo sueldo para todos los clientes que lo tengan. Indica la DF que provoca redundancia. ¿Es necesaria corregirla? ¿Cómo la corriges? Define en SQL lo necesario.

Método de Razonamiento:

1. Tengo la DF Título -> Sueldo. ¿Es necesario corregirla?
 2. Solo si es una DF mala. ¿Cómo se si es buena o mala?
 3. Si ha quedado en 3ªFN o superior es buena. ¿Cómo se si está en 3ª FN?
 4. Porque cumple los axiomas de 3ªFN. ¿Las cumple?
 - 4.1. Sí. Por lo que entonces está en 3ªFN y puedo dejarla porque la DF es buena.
 - 4.2. No. La DF es mala, debo corregirla. ¿Cómo?
 - 4.2.1 Normalizando. ¿Cómo?
- Explicación completa:

Es necesaria corregirla porque hay una DF transitiva (DNI -> Título -> Sueldo) luego se queda en 2ª FN y debemos tener como mínimo 3ªFN para una calidad razonable.

Se puede corregir normalizando, descomponiendo en dos tablas T1(DNI, Título) y T2(Título, sueldo) asumiendo que el cliente tiene solo un Título.

En desnormalización se puede considerar no descomponer por ser solo un atributo, y controlar la redundancia con un trigger que se activa cada vez que alguien cambia el sueldo a las cajas.

b.4) Si, en la tabla Puesto, ahora suponemos que un cliente tiene dos Títulos, ¿Cuál debe ser la CP? Corrige lo necesario en la BD dentro de Oracle y añade filas con datos inventados para que cumpla lo dicho.

Para tener identificación única de una fila necesito que la CP sean DNI y Título juntos. Sólo así todos los atributos dependen funcionalmente de la clave.

```
-- Borramos la tabla

DROP TABLE PUESTO;

-- Puesto: P(DNI, Título, Sueldo)

CREATE TABLE Puesto(DNI CHAR(8) not null,
Título VARCHAR2(30),
Sueldo FLOAT,
PRIMARY KEY (DNI, Título));

INSERT INTO Puesto VALUES ('00000001', 'cajera', 300);
INSERT INTO Puesto VALUES ('00000001', 'estudiante', 301);
INSERT INTO Puesto VALUES ('00000002', 'estudiante', 301);
INSERT INTO Puesto VALUES ('00000003', 'Presidente', 30000);
INSERT INTO Puesto VALUES ('00000004', 'VicePresidente', 3000);
INSERT INTO Puesto VALUES ('00000005', 'Presidente', 30000);
INSERT INTO Puesto VALUES ('00000006', 'Parado', 0);
```

c) Qué relaciones se tienen que dar (en cuanto al significado de los atributos) en cada apartado, para que exista la DF indicada:

c.1) EJEMPLO: Si se cumple en Empresa: Cotización -> Capital.

Cuando las cotizaciones de las empresas fijan el capital que tienen. Es decir, que para dos empresas con la misma Cotización tendrán el mismo Capital.

c.2) Tabla Cliente: NombreC -> DNI. Además de decir las relaciones: comprueba en las filas creadas si se cumplen esas relaciones y Corrige las que no las cumplan, inventando datos.

No puede haber dos clientes con el mismo nombre. Pero si los hay, las filas del cliente 1, 4, y 5 tienen el mismo nombre. Hay que inventar nombres únicos para cada cliente. (Si lo implemento puedo provocar esa DF poniendo el atributo NombreC como unique).

c.3) Tabla Compras: Tienda -> NumF Además de decir qué significa la DF, indica si es razonable esta DF y por qué.

Si cada Tienda tuviera un solo NumF, no es razonable que eso suceda.

c.4) Tabla Compras: NumF -> Tienda Además de decir las relaciones, indica si es razonable esta DF y por qué.

Un NumF corresponde a una tienda: solo si no hubiera, para dos tiendas, el mismo NumF, esto es, que la numeración de facturas fuera única para todas las tiendas. Esto no es razonable.

d) Descubrir la Dependencia y haz un procedimiento para mantener la consistencia de datos:

d.1) ¿Qué dependencia y de qué clase hay en la tabla Invierte, en la siguiente situación?:

- Cada Cliente invierte en varias Empresas, pero está obligado a invertir en los mismos Tipos en cada empresa que invierta; y cada Tipo debe tener la misma cantidad para cada empresa.

Esta situación provoca que:

- Cada vez que un Cliente invierte en una empresa nueva, tiene que invertir en todos los tipos en los que había invertido antes.

- También sucede que cada vez que el Cliente quiera invertir en otro Tipo diferente a los que ya tenía, le obligan a invertir en todas las empresas (en las que ya tiene inversiones) con ese Tipo y la misma cantidad.

Es una DM: DNI -> NombreE, DNI -> Tipo, cantidad.

Cuando un cliente decide invertir en un Tipo nuevo, hay que añadir tantas filas como Empresas en las que tiene inversiones. Y cuando se borra un Tipo hay que recorrer toda la tabla borrando las inversiones de ese Tipo. Si hay que borrar una Empresa, hay que borrar las inversiones de cada Tipo.

d.2) (mantengo desnormalizada) Implementa un procedimiento insertoConsistente, que vigile la consistencia de datos y que mantenga la dependencia automáticamente cada vez que se inserte una nueva inversión (será el parámetro del proc), manipulando las filas necesarias.

Obliga a tener filas para un cliente y empresa con todos los tipos. En la inserción de un Tipo nuevo, hay que insertar una fila en cada Empresa. Para eliminar un Tipo, hay que borrar varias filas.

d.3) (Normalizo la tabla) ¿Cómo podrías arreglar la dependencia del apartado anterior dividiendo en tablas?

Haciendo dos tablas según la DM:

1. Invierte-NombreE(DNI, NombreE)
2. Invierte-Tipo(DNI, Cantidad, Tipo)

Así ya se ve que, cada vez que invierta en una nueva empresa, solo inserta en Invierte-NombreE.

d.4) Nos dicen que el número de tarjeta NumT está compuesto por pais, banco, numLocal, así que compras quedaría: Compras: CO(DNI, NumF, Pais, Banco, NumLocal, Fecha, Tienda, Importe)

d.4.1) Describe ahora qué DFs hay ahora en esa tabla, justificando cada una con el significado que tiene.

```
create table Compras
(DNI          CHAR(8)  not null,
 Banco        CHAR(20) ,
 NumF         INT,
 Fecha       INT,
 Tienda       CHAR(20) ,
 Importe      INT,
 NumLocal     INT,
 Pais        CHAR(20) ,
 PRIMARY KEY (DNI, NumF, NumLocal),
 FOREIGN KEY (DNI) REFERENCES Cliente(DNI);
```

La clave primaria sería DNI, Pais, Banco, NumLocal y NumF. También tiene una clave extranjera que sería entre el cliente y DNI. Y se pierde la clave extranjera NumT con Tarjeta.

APARTADO 3. Formas Normales (FN): Reconocerlas, y pasar de una FN a la siguiente

Usando de nuevo la BDEjemplo original resuelve los siguientes apartados. Tendrás que determinar las claves primarias antes de hacer el apartado:

3.1 Queremos hacer que la tabla Cliente NO esté en la 1ª FN: (Antes indica cual es la CP) Qué relaciones deben cumplir los atributos para que suceda eso. Da ejemplos de filas.

Un cliente tenga varios teléfonos, así Teléfono sea multivalorado luego No está en 1ªFN.

3.2 Introduce una DF en la tabla Invierte para que NO esté en la 2ª FN: (Antes indica cual es la CP) ¿Qué relaciones deben cumplir los atributos para que suceda eso? (define la CP que usas). Da ejemplos de filas. ¿Cuál sería el arreglo?

La CP es: DNI, NombreE, Tipo. DF: Para un Tipo siempre es la misma Cantidad. Tipo -> Cantidad

No está en 2ªFN porque esta DF depende parcialmente de la CP.

Arreglo al hacer dos tablas:

1. Invierte-a(DNI, NombreE, Tipo)
2. Tipo-C(Tipo, Cantidad)

Desnormalización: Como es solo un atributo, no sería muy problemático dejar que se repitiera, si a cambio tenemos que acceder a dos tablas frecuentemente en lugar de una sola.

3.3 Introduce una DF en la tabla Empresa para que NO esté en la 3ª FN: (Antes indica cual es la CP) ¿Qué relaciones deben cumplir los atributos para que suceda eso? (define la CP que usas)

Da ejemplos de filas. ¿Cuál sería el arreglo?

La CP es NombreE. La DF: Cotización -> Capital siempre una cotización tiene el mismo capital. No está en 3ªFN porque hay una DF transitiva: NombreE → Cotización → Capital
Arreglo al hacer dos tablas:

1. Empresa-1(NombreE, Cotizacion)
2. CotizacionE (Cotizacion Capital)

Desnormalización: Como es solo un atributo, no sería muy problemático dejar que se repitiera, si a cambio tenemos que acceder a dos tablas frecuentemente en lugar de una sola.

3.4 Introduce una DF en la tabla Tarjeta para que NO esté en la FNBC: (Antes indica cual es la CP) ¿Qué relaciones deben cumplir los atributos para que suceda eso? (define la CP que usas). Da ejemplos de filas. ¿Cuál sería el arreglo?

La CP es NumT, Organizacion. La DF es TipoT -> Organizacion, no está en FNBC porque una parte de la clave Y depende de un atributo X no superclave.

Arreglo al hacer dos tablas, pero puede perder la DF que hacía la clave primaria, por lo que no es recomendable en este caso.