

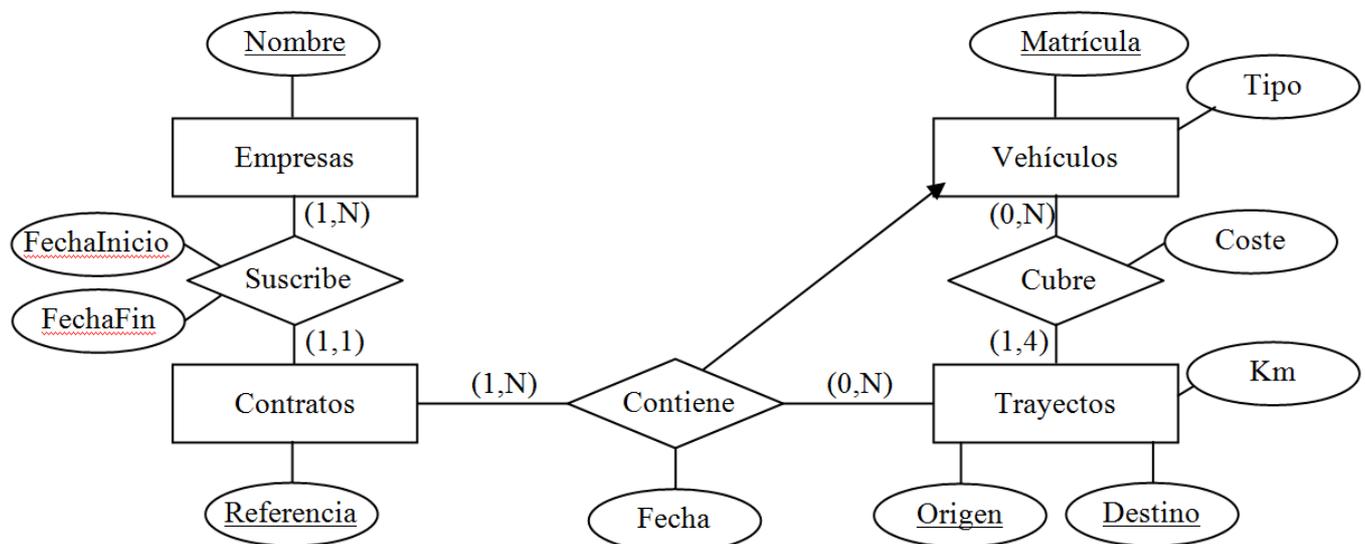
Examen de Bases de datos Grado de Ingeniería en Informática 6 de Febrero, 2014

1) (3 puntos) Una empresa de logística desea implementar una base de datos relacional que cumpla los siguientes requisitos:

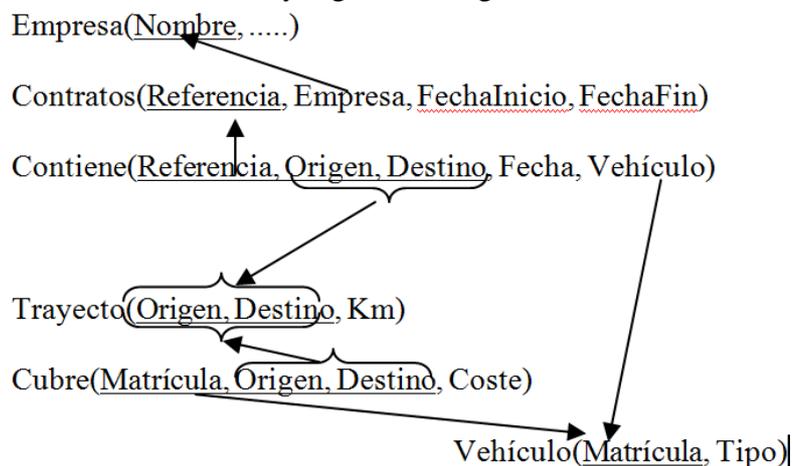
La empresa logística ofrece sus servicios de transporte a diferentes empresas bajo un contrato que se suscribe durante un periodo de tiempo. Un contrato se identifica por su referencia, y puede incluir varios trayectos (al menos uno), indicando la fecha de cada trayecto y el vehículo concreto que lo va a realizar. Los trayectos son identificados por un origen y un destino y se conoce el número de Km. La empresa de logística cuenta con vehículos de transporte que pueden cubrir determinados trayectos. Un vehículo tiene una matrícula y un tipo (marca). Cada trayecto está cubierto por al menos un vehículo y como mucho por cuatro vehículos distintos y su coste viene determinado por el vehículo que lo recorra.

- (1,5 puntos) Se pide diseñar el esquema conceptual con el modelo entidad-relación especificando las restricciones de participación mínimo-máximo.
- (1 puntos) Traducir dicho modelo a modelo relacional e incluir las reglas de Integridad referencial
- (0,5 puntos) Indicar las restricciones no reflejadas en el modelo relacional

Solución:



El modelo relacional y reglas de integridad referencial son:





Las restricciones no reflejadas en el modelo relacional son:

-Un trayecto es cubierto por hasta 4 vehículos

Restricciones de participación total en un M a M

- Un contrato contiene al menos un trayecto

- Un trayecto viene cubierto por al menos un vehículo.

2) (2 puntos) Dada las relaciones (no necesariamente del modelo anterior):

Contratos(Referencia, Empresa, FechaInicio)

Contiene(Referencia, Origen, Destino, Fecha, Vehículo, coste)

Escribir en SQL las siguientes consultas:

- (0,25 punto) Mostrar en orden alfabético los vehículos que se usan en trayectos contratados por empresas que empiezan por 'A'

```
SELECT DISTINCT Vehiculo
FROM Contratos, Contiene
WHERE Contratos.Referencia = Contiene.Referencia
      AND Empresa IS LIKE 'A%'
ORDER BY Vehiculo
```

- (0,25 punto) Consultar con operaciones de álgebra relacional los vehículos usados en trayectos contratados por IBM

$\Pi_{\text{Vehículo}} (\sigma_{\text{Empresa}='IBM'} (\text{Contratos} \bowtie \text{Contiene}))$

- (0,5 punto) Mostrar todas las empresas con más de 5 contratos iniciados en 2014

```
SELECT Empresas
FROM Contratos
WHERE FechaInicio > '01.01.2014'
GROUP BY Empresa
HAVING COUNT (*) > 5
```

- (0,5 puntos) Mostrar el vehículo más barato para el trayecto Madrid-Barcelona

```
SELECT DISTINCT Vehiculo
FROM Contiene
WHERE Origen = 'Madrid'
      AND Destino = 'Barcelona'
AND Coste = (SELECT MIN(Coste)
              FROM Contiene
              WHERE Origen = 'Madrid'
                AND Destino = 'Barcelona'
            )
```

- (0,5 puntos) Borrar los contratos que no tienen ningún Trayecto incluido

```
DELETE FROM Contratos C
WHERE NOT EXISTS (SELECT *
                  FROM Contiene
                  WHERE Referencia = C.Referencia)
```



3) (2,5 puntos) Dado el modelo (no necesariamente de ejercicios anteriores)

Contratos(Referencia, Empresa, Fecha)

Contiene(Referencia, Origen, Destino, Vehículo)

Se modifica el diseño de Contratos añadiendo un atributo redundante que represente el número de trayectos contratados para cada contrato:

Contratos(Referencia, Empresa, Fecha, NumTrayectos)

- (0.5 puntos) Crear con SQL DDL ambas tablas incluyendo la nueva columna NumTrayectos en la tabla de Contratos e implementando Integridad Referencial de manera que si se borra un contrato se debe borrar toda su información.
- (1 punto) Escribir un procedimiento almacenado con una referencia de contrato como parámetro de entrada (CHAR(6)) y actualice su contador de trayectos y lo imprima por consola. Se debe declarar una excepción que se lance para dar un mensaje si la referencia no tiene trayectos.

```
CREATE OR REPLACE PROCEDURE NumTrayectos (IN numRef CHAR(6)) IS
cont NUMBER;
noTrayectos EXCEPTION;
BEGIN

SELECT COUNT(*)
INTO cont
FROM Contiene
WHERE Referencia = numRef;

UPDATE Contratos
SET NumTrayectos = cont
WHERE Referencia = numRef;

IF cont = 0 THEN RAISE noTrayectos
END IF;

DBMS_OUTPUT.PUT_LINE(' El contrato ' || numRef || ' tiene ' || cont || ' trayectos');

EXCEPTION
WHEN noTrayectos THEN DBMS_OUTPUT.PUT_LINE(' El contrato ' || numRef || ' no tiene
trayectos');
END;
```

- (1 punto) Crear un disparador (trigger) que mantenga actualizado el atributo redundante al insertar y borrar en la tabla.

```
create or replace TRIGGER NumTr
BEFORE INSERT OR DELETE
ON Contiene
FOR EACH ROW
BEGIN
IF INSERTING THEN
UPDATE Contratos
SET NumTrayectos = NumTrayectos + 1
WHERE Referencia = :NEW.Referencia;
END IF;

IF DELETING THEN
UPDATE Contratos
SET NumTrayectos = NumTrayectos -1
WHERE Referencia = :OLD.Referencia;
END IF;
END;
```



4) (1 puntos) Describir los niveles de aislamiento (ISOLATION) de Oracle (SERIALIZABLE y READ COMMITED).

5) (1,5 puntos) Dado el modelo (no necesariamente el de ejercicios anteriores)

Contratos(Referencia, Empresa, NIFEmpresa, fecha)

Sabiendo que entre Empresa y NIFEmpresa hay una correspondencia uno a uno, y que una empresa sólo puede firmar un contrato al día:

- (0,5 puntos) Dar un recubrimiento mínimo de dependencias funcionales

$F = \{$ Referencia \rightarrow Empresa,
Referencia \rightarrow Fecha,
Empresa \rightarrow NIFEmpresa,
NIFEmpresa \rightarrow Empresa,
Empresa, fecha \rightarrow Referencia $\}$

- (0,5 puntos) Listar las claves candidatas

Las claves candidatas son {Referencia} y {Empresa, fecha}

- (0,5 puntos) Descomponer Contratos hasta la 3FN

Contratos(Referencia, Empresa, fecha)

Empresa(Empresa, NIFEmpresa)

O bien

Contratos(Referencia, Empresa, fecha)

Empresa(NIFEmpresa, Empresa,)