

Fundamentos de Bases de Datos. Ordinaria 2018

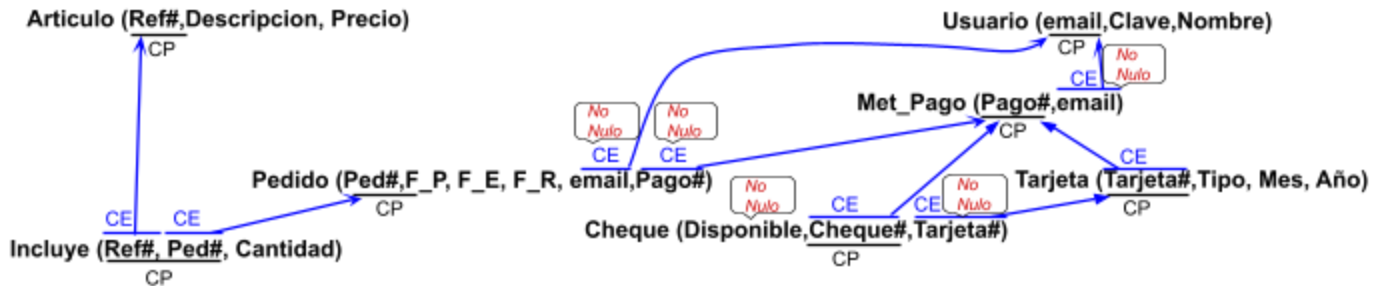
Parcial 2. Ejercicio Práctico

Apellidos y Nombre:

Grupo:

Profesor del Grupo:

Disponemos de la siguiente BD sobre gestión de pedidos por Internet.



1. Escribe las instrucciones en SQL para la creación de las tablas PEDIDO y CHEQUE. Se presuponen creadas todas las restantes tablas. Además de las restricciones de integridad especificadas en el dibujo, deben considerarse las siguientes:
 - a. Ped# es una cadena de hasta 8 caracteres. Disponible no puede tomar valores negativos.
 - b. F_P, F_E y F_R, representan la fecha de pedido, la de envío y la de recepción, respectivamente y son de tipo fecha. No existen pedidos anteriores al año 2010. Debe garantizarse además que $F_P \leq F_E \leq F_R$.
 - c. No puede cargarse un Cheque# a una Tarjeta#, teniendo ambas con la misma identificación.

(1.5 pts).

```
CREATE TABLE PEDIDO (
  Ped# varchar2(8) PRIMARY KEY,
  F_P date CHECK (To_number(to_char(F_P, 'YYYY')) > 2009),
  F_E date, F_R date,
  email NOT NULL REFERENCES usuario(email),
  Pago# NOT NULL REFERENCES Met_Pago(pago#),
  CHECK (F_P <= F_E AND F_E <= F_R));
```

```
CREATE TABLE CHEQUE (
  Cheque# PRIMARY KEY REFERENCES Met_Pago(Pago#),
  Tarjeta# NOT NULL REFERENCES Tarjeta(Tarjeta#),
  Disponible NUMBER CHECK (Disponible >= 0),
  CHECK (Cheque# <> Tarjeta#));
```

2. Mostrar los nombres de los usuarios que hayan pedido todos los artículos cuyo precio unitario es inferior a 20e. **(AR 1.25 ptos, CRT 1.5 ptos, SQL 1.25 ptos)** .

$$\Pi_{nombre} (Usuario \bowtie (\Pi_{email, ref\#} (Pedido \bowtie Incluye) \div \Pi_{ref\#} (\sigma_{Precio < 20} (Articulo))))$$

$$\{ U.nombre \mid usuario(U) \text{ and not } (exists A)(articulo(A) \text{ and } A.precio < 20 \text{ and not } (exists P, I)(pedido(P) \text{ and incluye(I) and } I.ref\# = A.ref\# \text{ and } I.ped\# = P.ped\# \text{ and } P.email = U.email))\};$$

RANGE U IN usuario; RANGE A IN articulo; RANGE I IN incluye ; RANGE P IN pedido

SELECT U.nombre WHERE $\neg \exists A($
 $A.precio < 20 \wedge \neg \exists I, P(I.ref\# = A.ref\# \wedge I.ped\# = P.ped\# \wedge P.email = U.email))$
 SELECT U.nombre WHERE $\forall A($
 $A.precio < 20 \rightarrow (\exists I, P(I.ref\# = A.ref\# \wedge I.ped\# = P.ped\# \wedge P.email = U.email))$

```
SELECT U.nombre FROM usuario U WHERE NOT EXISTS (
    SELECT * FROM articulo A WHERE A.precio < 20 AND NOT EXISTS (
        SELECT * FROM pedido P, incluye I WHERE I.ref# = A.ref#
        AND I.ped# = P.ped# AND P.email = U.email));
```

ó

```
SELECT U.nombre FROM usuario U WHERE NOT EXISTS (
    SELECT A.ref# FROM articulo A WHERE A.precio < 20
    MINUS
    SELECT I.ref# FROM pedido P, incluye I WHERE
        AND I.ped# = P.ped# AND P.email = U.email);
```

3. Mostrar el código del pedido junto con su importe total, de aquellos pedidos cuyo importe total sea mayor que 1000 e, ordenados por importe total. **(SQL 1.75 ptos)**

```
SELECT P.Ped#, SUM(P.precio * P.cantidad)
FROM (incluye NATURAL JOIN articulo) P
GROUP BY P.ped# HAVING SUM(P.precio * P.cantidad) > 1000
ORDER BY SUM(P.precio * P.cantidad);
```

4. Mostrar los datos del pedido más antiguo. **(AR 1.5 ptos, CRT 1.25 ptos)**

$\rho(pedido) = P1, P2$

$$(Pedido \bowtie (\Pi_{ped\#} (P1) - \Pi_{P1.ped\#} (\sigma_{P1.F_P > P2.F_P} (P1 \times P2))))$$

$$\{ P \mid pedido(P1) \text{ and not } (exists P2)(pedido(P2) \text{ and } P1.F_P > P2.F_P) \};$$

$$\{ P \mid pedido(P1) \text{ and } (forall P2)(pedido(P2) \text{ and } P1.F_P \leq P2.F_P) \};$$

RANGE P1, P2 IN pedido

SELECT P1.* WHERE $\neg \exists P2(P1.F_P > P2.F_P)$

SELECT P1.* WHERE $\forall P2(P1.F_P \leq P2.F_P)$

Tiempo total de realización (Teoría y Prácticas del Segundo Parcial): **1 hora y 30 minutos.**