

## Fundamentos de Bases de Datos

### Parcial 2. Ejercicio Práctico

Apellidos y Nombre:

Grupo:

Profesor del Grupo:

Disponemos de la siguiente BD que gestiona los ordenadores de una empresa:



Se pide:

- A. Escribe las instrucciones en SQL para la **creación** de la tabla REPARA y la **inserción** de una tupla en dicha tabla (1.5 pts.). Para la creación de la tabla se presuponen creadas el resto de tablas. Además de las restricciones de integridad especificadas en el esquema, deben considerarse las siguientes:
- El costo debe ser un valor positivo y no debe superar los 5000€.
  - La fecha de inicio no puede ser superior a la fecha de finalización.
  - Por defecto la causa será 'desconocido'.

```
CREATE TABLE repara (
  ID REFERENCES ordenador(ID) ,
  F_ini DATE,
  NIF REFERENCES empresa(NIF) ,
  F_fin DATE,
  Causa VARCHAR2(100) DEFAULT 'desconocido',
  Costo NUMBER CHECK(costo BETWEEN 1 AND 5000),
  CHECK(f_ini<=f_fin)
  PRIMARY KEY (ID,F_ini))
;
```

```
INSERT INTO repara VALUES(30,sysdate,'2222222G',sysdate+1,'Fuente
alimentación',100);
```

- B. Realiza las siguientes consultas:

- a. Listar los nombres de aquellos usuarios a los que le han tenido que reparar todos los ordenadores que han tenido asignados en algún momento (en CRT 1.5 pts. y SQL 1.25 pts.).

```
{ U.nombre | usuario(U) and not (exists A) (asigna(A) and A.DNI=U.DNI and not (exists R) (repara(R) and R.ID=A.ID))};
```

o

```
{ U.nombre | usuario(U) and (forall A) (asigna(A) and A.DNI<>U.DNI or (exists R) (repara(R) and R.ID=A.ID))};
```

```
SELECT U.nombre FROM usuario U WHERE NOT EXISTS(
    SELECT * FROM asigna A WHERE A.DNI=U.DNI AND NOT EXISTS (
        SELECT * FROM repara R WHERE R.ID=A.ID
    ))
```

o

```
SELECT U.nombre FROM usuario U WHERE NOT EXISTS(
    SELECT A.ID FROM asigna A WHERE A.DNI=U.DNI
    MINUS
    SELECT R.ID FROM repara r
)
```

- b. Encontrar todos los usuarios (mostrar sólo el nombre y su fecha de nacimiento) que tienen o han tenido asignados al menos dos ordenadores. (en AR 1.5 pts.).

$\rho(\text{Asigna}) = a1, a2$

$\pi_{\text{nombre}, f\_nac}(\text{Usuario} \bowtie \pi_{a1.DNI}(\sigma_{a1.DNI=a2.DNI \wedge a1.ID < a2.ID}(a1 \times a2)))$

```
SELECT nombre, f_nac FROM usuario NATURAL JOIN asigna
GROUP BY (nombre, f_nac) HAVING count(*) >1;
```

$\rho(\text{Asigna} \bowtie (\pi_{ID}(\sigma_{\text{marca}='Acme'}(\text{ordenador}))) = a1, a2$

$\pi_{\text{nombre}, f\_nac}(\text{Usuario} \bowtie \pi_{a1.DNI}(\sigma_{a1.DNI=a2.DNI \wedge a1.ID < a2.ID}(a1 \times a2)))$

```
SELECT nombre, f_nac FROM usuario NATURAL JOIN asigna NATURAL JOIN
(select ID from ordenador WHERE marca='Acme')
GROUP BY (nombre, f_nac) HAVING count(*) >1;
```

- c. Mostrar los identificadores de los ordenadores portátiles que aún no han sido reparados por la empresa 'ReparaINC' (en **AR 1.25pts.** y **CRT 1.5 pts.**).

$$\pi_{ID}(\sigma_{\text{tipo}='Portatil'}(\text{Ordenador})) - \pi_{ID}(\text{Repara} \bowtie \sigma_{\text{nombre}='ReparaINC'}(\text{Empresa}))$$

```
{ O.ID | ordenador(O) and O.tipo='Portatil' and not (exists E,R) (empresa(E)
and repara(R) and E.nombre='ReparaINC' and R.NIF=E.NIF and R.ID=O.ID) };
```

- C. Crear una vista que muestre **aquellos** ordenadores que han tenido que ser reparados más de 5 veces o cuyo costo total de reparación hasta el momento supere los 500€, la vista deberá incluir los datos de los ordenadores. Ordene los resultados por marca y modelo (**1.5 pt.**).

La solución sería:

```
CREATE VIEW vista AS SELECT ordenador.ID, marca, modelo, tipo, FROM
ordenador, repara
WHERE
ordenador.ID= repara.ID and (5<(SELECT count(*) FROM repara re
WHERE ordenador.ID=re.ID) or 500<(SELECT sum(costos) FROM repara rt
WHERE ordenador.ID=rt.ID)) GROUP BY ordenador.ID,marca, modelo, tipo
ORDER BY marca,modelo
```

ó

```
CREATE VIEW vista AS SELECT O.ID, marca, modelo, tipo FROM ordenador O,
repara R
WHERE O.ID= R.ID
GROUP BY (O.ID, marca, modelo, tipo)
HAVING count(*) >5 OR sum(costo) >500
ORDER BY marca,modelo
```

o

```
CREATE VIEW vista AS SELECT ID, marca, modelo, tipo FROM ordenador
WHERE ID IN
(SELECT ID FROM repara R
GROUP BY ID
HAVING count(*) >5 OR sum(costo) >500)
ORDER BY marca,modelo
```

Tiempo total de realización (Teoría y práctico): 1.30h.