

Parcial2-Practico-Ordinaria2020-...



EasyIngeniero



Fundamentos de Bases de Datos



2º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación
Universidad de Granada



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.



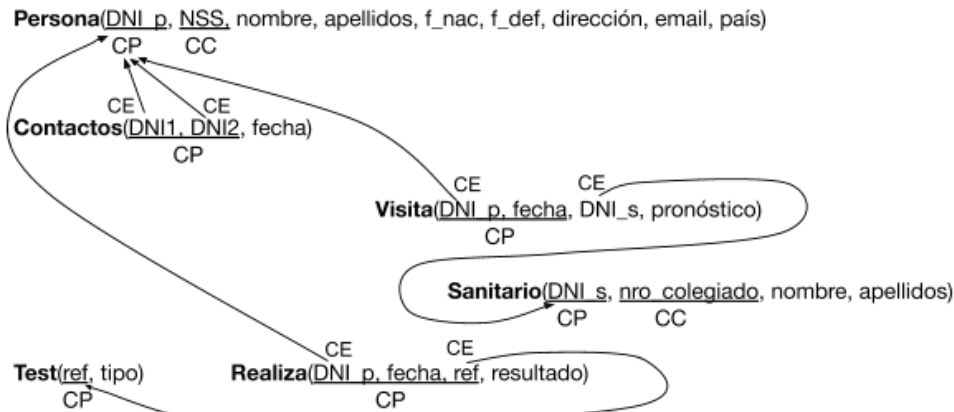


Descarga la APP de Wuolah.

Ya disponible para el móvil y la tablet.



Dadas las siguientes tablas:



Resuelva las siguientes tareas:

A) Cree la tabla Persona y la tabla Visita suponiendo creadas todas las demás tablas. Asuma las restricciones del gráfico anterior conjuntamente con las siguientes **(1.75 puntos)**:

- Todos los atributos de la tabla Persona deben ser no nulos con excepción del email.
- El país por defecto será "España".
- El Número de la Seguridad Social (NSS) está compuesto por nueve dígitos.
- El atributo f_def tomará por defecto la fecha: 1 de enero de 3000.
- El pronóstico guardará un texto plano que puede llegar a ser potencialmente extenso.

SOL:

```
CREATE TABLE persona (
```

```
DNI_p varchar2(9) PRIMARY KEY, NSS NUMBER(9) UNIQUE NOT NULL, Nombre
varchar2(30) NOT NULL, f_nac DATE NOT NULL, f_def DATE DEFAULT
TO_DATE('01/01/3000','DD/MM/YYYY'), Dirección varchar2(50) NOT NULL,
email varchar2(50), Pais varchar2(25) NOT NULL DEFAULT 'España');
```

```
CREATE TABLE visita (DNI_p REFERENCES persona(DNI_p), fecha DATE, DNI_s
REFERENCES sanitario(DNI_s), prognosis LONG , PRIMARY KEY (DNI_s, fecha));
```

B) Resuelva las siguientes consultas:

1. Muestre los nombres y apellidos del "caso cero", es decir, aquella persona que ha fallecido antes que todas las demás (en **Álgebra Relacional 1.25 puntos**.)

$$\rho(persona) = p1, p2$$

$$\Pi_{nombre, apellidos}(\Pi_{DNI_s, nombre, apellidos}(Persona) - \Pi_{p1.DNI_s, p1.nombre, p1.apellidos}(\sigma_{p1.fdef > p2.fdef}(p1 \times p2)))$$

2. Mostrar el DNI de aquellas personas que han sido atendidas por todos los sanitarios antes de marzo del 2020. (en **Álgebra Relacional 1.25 puntos** y en **SQL 1.25 puntos**)

$$\Pi_{DNI_p, DNI_s}(\sigma_{fecha < '01/03/2020'}(visita)) \div \Pi_{DNI_s}(sanitario)$$

```
SELECT p1.DNI_p FROM persona p1 WHERE NOT EXISTS (
SELECT DNI_S FROM sanitario
MINUS
SELECT v.DNI_S FROM visita v WHERE v.fecha <
TO_DATE('01/03/2020', 'dd/mm/yyyy') AND v.DNI_p=p1.DNI_p);
```

3. Mostrar el DNI de las personas a las que sólo le han realizado tests de tipo "PCR". (en **Álgebra Relacional 1.25 puntos**)

$$\Pi_{DNI_p}(\sigma_{tipo='PCR'}(realiza \bowtie test)) - \Pi_{DNI_p}(\sigma_{tipo \neq 'PCR'}(realiza \bowtie test)) \quad \text{ó}$$

$$\Pi_{DNI_p}(realiza) - \Pi_{DNI_p}(\sigma_{tipo \neq 'PCR'}(realiza \bowtie test))$$

C) Crear una vista que informe de los datos de contacto (nombre, apellidos, dirección e email) de los potenciales "supercontagadores", personas que aún siguen vivas y que han estado en contacto con al menos otras 50 personas. Podemos suponer que si una persona (P1) ha entrado en contacto con otra persona (P2) aparecerá en la tabla de contactos dos veces: como (P1,P2,fecha) y como (P2,P1,fecha). (1.5 puntos)

```
CREATE VIEW supercontagadores AS
SELECT nombre, apellidos, dirección, email FROM persona WHERE DNI_p IN
(SELECT DNI1 FROM contactos
WHERE f_def=TO_DATE('01/01/3000', 'DD/MM/YYYY')
GROUP BY (DNI1) HAVING COUNT(*) >=50);
-----
SELECT nombre, apellidos, dirección, email FROM persona WHERE
f_def=>SYSDATE AND (SELECT COUNT(*) FROM contactos WHERE
contactos.DNI1=PERSONA.DNI_p)>50;
```

D) Modificar la tabla Persona para que incorpore un atributo llamado grupo_de_riesgo y actualice la tabla para que este atributo tome el valor 'SI' cuando la persona esté en un grupo de riesgo, es decir, tenga 70 años o más. (1.75 puntos) (1.75 puntos)

```
ALTER TABLE persona ADD ( grupo_de_riesgo varchar2(2) CHECK (grupo_de  
riesgo='SI' OR grupo_de_riesgo='NO'));
```

```
UPDATE persona SET grupo_de_riesgo='SI' WHERE (sysdate-f_nac)/365.25<=70;
```