

# **Simulacro 2 parcial**

Curso Académico 2019/2020

Disponemos de la siguiente BD que gestiona los re

**USUARIO**(ID, nombre, apellido, gér

**TEST**(REF, tipo)

**PREGUNTA**(REF, NRO, texto, rta\_correcta)

**RESPONDE**(ID, REF, NRO, rta\_usuario, tiempo, fecha)

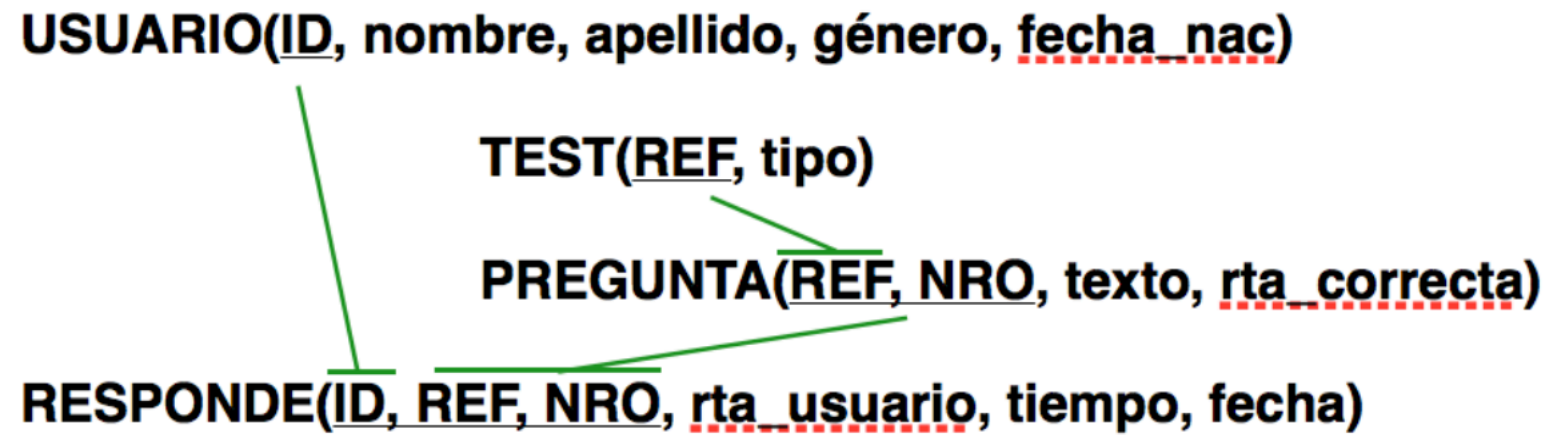
Se pide:

- Escribe las instrucciones en SQL para la creación de la tabla RESPONDE y la inserción de una tupla en dicha tabla. 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:
  - **rta\_usuario** indica la respuesta (V o F) que ha dado el usuario a esa pregunta;
  - **tiempo** indica el tiempo en segundos que ha tardado un usuario en responder la pregunta, el tiempo máximo permitido es de 5 minutos;
  - **fecha** corresponde a la fecha en que se ha respondido a la pregunta, por defecto será la fecha actual;
  - no puede haber ningún valor nulo

```
CREATE TABLE Responde (  
  ID CHAR(5) REFERENCES Usuario(ID),  
  REF CHAR(3),  
  NRO INT,  
  rta_usuario CHAR(1) NOT NULL CHECK (rta_usuario IN ('V','F')),  
  tiempo INT NOT NULL CHECK (tiempo BETWEEN 0 AND 300),  
  fecha DATE NOT NULL DEFAULT SYSDATE,  
  PRIMARY KEY (ID, REF, NRO),  
  FOREIGN KEY (REF, NRO) REFERENCES Pregunta(REF, NRO)  
);
```

```
INSERT INTO Responde VALUES ('YO', 'X', 1, 'V', 60, SYSDATE);
```

- Crear una vista que muestre la media y la desviación estándar del tiempo empleado en cada test, pero solamente de aquellos tests que hayan sido realizados por al menos 10 usuarios.



```
CREATE VIEW Vista AS
(SELECT REF, AVG(tiempo), STDDEV(tiempo)
FROM Responde
GROUP BY REF
HAVING COUNT(DISTINCT ID) >= 10);
```

- Realiza las siguientes consultas en AR:

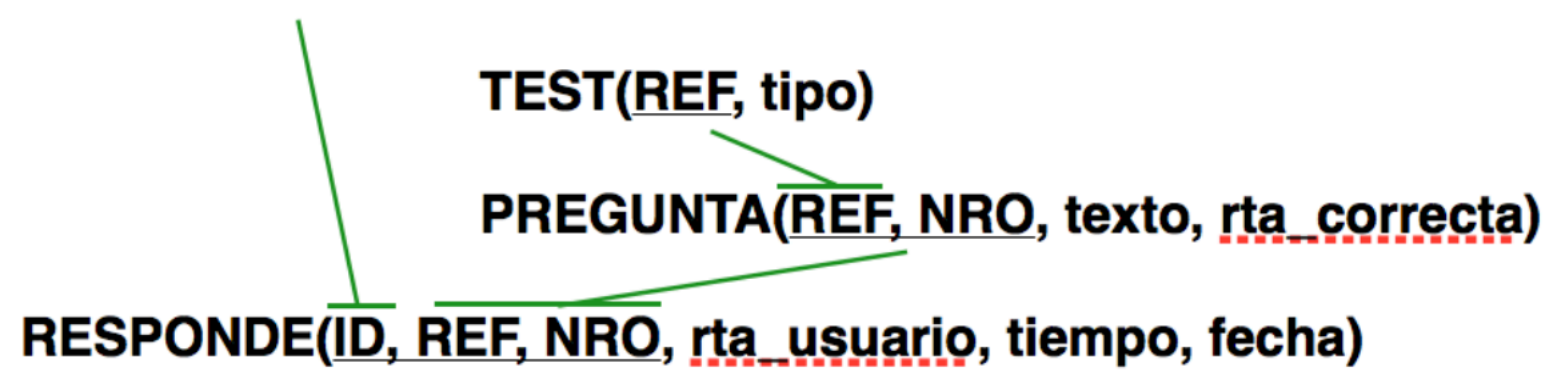
1. Listar los identificadores de aquellas preguntas que no han sido acertadas por ningún usuario.

**USUARIO**(ID, nombre, apellido, género, fecha\_nac)

**TEST**(REF, tipo)

**PREGUNTA**(REF, NRO, texto, rta\_correcta)

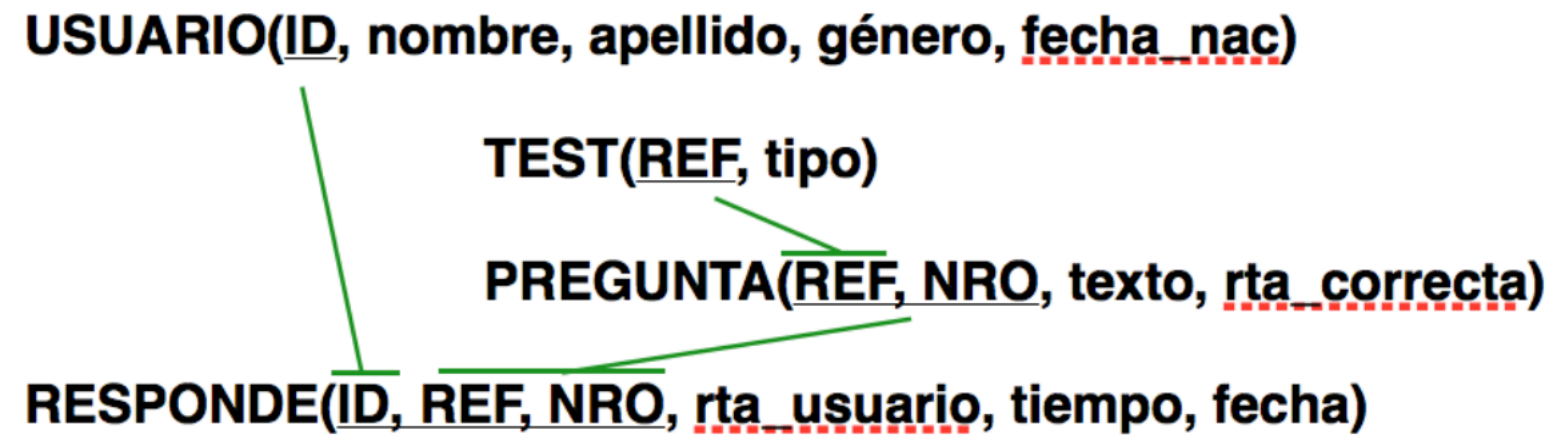
**RESPONDE**(ID, REF, NRO, rta\_usuario, tiempo, fecha)



$$\pi_{REF,NRO}(Pregunta) - \pi_{REF,NRO}(\sigma_{rta\_usuario=rta\_correcta}(Pregunta \bowtie Responde))$$

- Realiza las siguientes consultas en AR:

2. Mostrar todos los datos de aquellas usuarias que han tardado más en responder una pregunta (sin importar si han respondido bien o mal).



$$\rho(\pi_{ID}(\sigma_{genero='F'}(Usuario)) \bowtie Responde) = RUA$$

$$\rho(RUA) = RUB$$

$$Usuario \bowtie (\pi_{ID}(\sigma_{genero='F'}(Usuario)) - \pi_{RUA.ID}(\sigma_{RUA.tiempo < RUB.tiempo}(RUA \times RUB)))$$

- Realiza las siguientes consultas en AR y SQL:

3. Mostrar el identificador de los usuarios que han respondido a todas las preguntas de la base de datos en menos de 1 minuto (sin importar el test o si han respondido bien o mal).

**USUARIO**(ID, nombre, apellido, género, fecha\_nac)

**TEST**(REF, tipo)

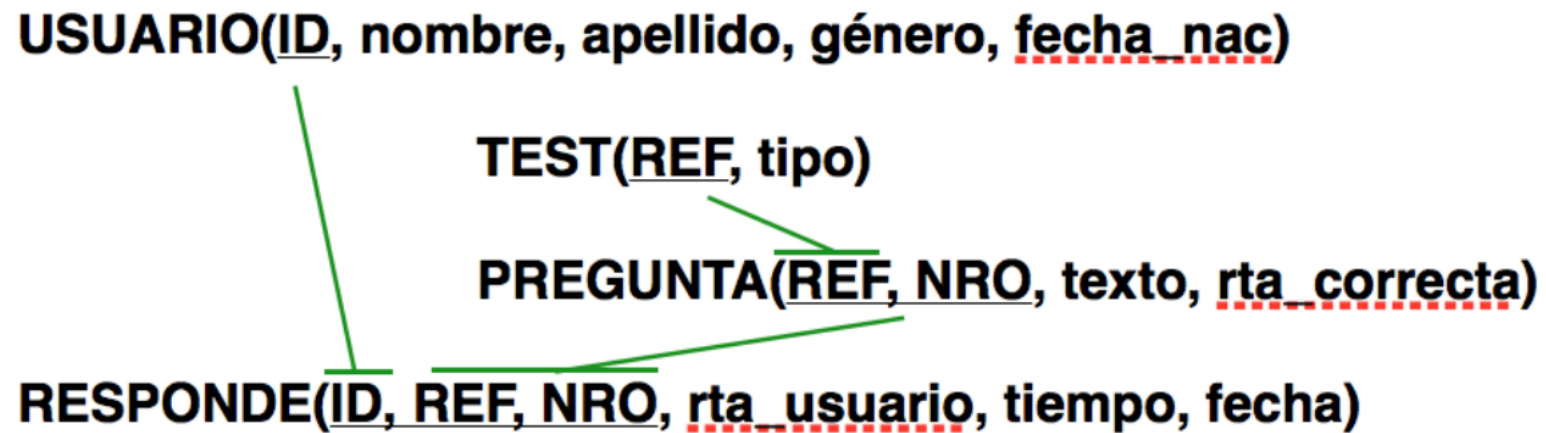
**PREGUNTA**(REF, NRO, texto, rta\_correcta)

**RESPONDE**(ID, REF, NRO, rta\_usuario, tiempo, fecha)

$$\pi_{ID,REF,NRO}(\sigma_{tiempo < 60}(Responde)) \div \pi_{REF,NRO}(Pregunta)$$

```
SELECT ID FROM Usuario WHERE NOT EXISTS (
  (SELECT REF,NRO FROM Pregunta)
  MINUS
  (SELECT REF,NRO FROM Responde WHERE (Responde.ID=Usuario.ID) AND
    (tiempo < 60)
  )
);
```

- Modifique el sistema de tablas para poder incorporar el promedio de notas de los tests de cada usuario y actualice los valores correspondientes. Para calcular la nota: cada respuesta correcta suma 1 y las incorrectas no restan.



```
ALTER TABLE Usuario ADD (promedio FLOAT);
```

```
UPDATE Usuario SET promedio =
(SELECT AVG(nota) FROM
(SELECT REF, COUNT(*) nota
FROM Responde NATURAL JOIN Pregunta
WHERE (Usuario.ID=Responde.ID) AND (rta_usuario=rta_correcta)
GROUP BY REF));
```

“Quiero calcular la nota del usuario a actualizar, por cada test y luego sacarle el promedio”