

B DATOS 2P 2019 - SQL - 14/05/2019

a) Cambiar todas las medidas con estado "incompleta" de los sensores del módulo "MA", tomadas en la calle Uría después de las 17:00, para que estado sea "OK"

```
UPDATE medida
SET estado = 'OK'
WHERE hora > '17:00' AND id-modulo = 'MA'
      id-modulo IN (SELECT id-modulo FROM modulo WHERE nom-mod = 'MA')
AND id-ubicacion IN (SELECT id-ubicacion FROM ubicacion
AND estado = 'incompleta' WHERE calle = 'Uría');
```

b) Identificación y descripción de los sensores que tengan el mayor número de medidas realizadas dentro de cada tipo de sensores. (imprimir también el tipo)

```
SELECT m.id-modulo, m.id-sensor, s.tipo
FROM medida AS m NATURAL JOIN sensor AS s
GROUP BY m.id-modulo, s.tipo,
HAVING COUNT(*) > ALL (SELECT COUNT(*)
FROM medida NATURAL JOIN sensor AS s2
WHERE s2.tipo = s.tipo
GROUP BY id-sensor, id-modulo)
```

c) Calles en las que nunca se ha realizado una medida de tipo "temperatura" con resultado "fallo". También hacer en álgebra relacional

```
(SELECT calle
FROM ubicacion AS u)
EXCEPT
(SELECT calle
FROM ubicacion NATURAL JOIN medida NATURAL JOIN sensor
WHERE tipo = "temperatura"
AND estado = "fallo")
```

← también que al menos tenga una medida de temperatura

$\pi_{calle}(u)$

d) Identificación del módulo y nombre del responsable, de aquellos módulos tales que todas las medidas que han realizado sus sensores de tipo 'humedad' han tenido valor > 100

```
SELECT t.id-modulo, n-resp.
FROM modulo AS t
WHERE (SELECT id-mod, id-sens, id-ubic, fecha, hora
FROM medida AS m
WHERE m.id-mod = t.id-mod AND
valor > 100 (AND tipo = 'humedad')
también vale
```

medidas del t.modulo > 100

CONTAINS

medidas del t.modulo de tipo 'humedad'

CONTAINS

```
(SELECT id-mod, id-sens, id-ubic, fecha, hora
FROM medida AS m NAT JOIN sensor AS s
WHERE m.id-mod = t.id-mod AND
s.tipo = 'humedad')
```